

SAULO BAZZI OBERDERFER

**ENGENHARIA DE WEBAPP PARA UM GRUPO
ESCOTEIRO UTILIZANDO O MÉTODO OOHDM**

Monografia apresentada à UNOCHAPECÓ como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Orientador: Jean Carlos Hennrichs

ENGENHARIA DE WEBAPP PARA UM GRUPO ESCOTEIRO

UTILIZANDO O MÉTODO OOHDm

SAULO BAZZI OBERDERFER

Esta Monografia foi julgada para obtenção do título de Bacharel em Ciência da Computação,
na área de Engenharia de Software e aprovada pelo curso de Ciência da Computação

ORIENTADOR: Prof. JEAN CARLOS HENNICHs

COORDENADOR(A) DO CURSO: Prof. Mônica Tissiani De Toni Pereira

BANCA EXAMINADORA

PRESIDENTE: Prof. Jean Carlos Hennrichs

Prof. Jusane Farina Lara

Prof. Saulo Popov Zambiasi

*“Procurem deixar este mundo um pouco melhor do que o encontraram
e quando chegar sua vez de morrer, poderão morrer felizes sentindo que pelo menos não
desperdiçaram seu tempo e fizeram o seu melhor possível”*

Baden-Powell

DEDICATÓRIA

Dedico este trabalho aos meus pais Vilson e Delise
por serem o meu maior e melhor exemplo de vida,
e as duas mulheres da minha vida Lara e Sahra
por me amarem mais do que o impossível.

AGRADECIMENTOS

Agradeço a todos os que foram meus professores, formal ou informalmente, que ao longo de minha vida procuraram transmitir seus conhecimentos, dividindo-os com honestidade e determinação.

Ao professor Jean Carlos Hennrichs tanto pelo incentivo quanto pela orientação, também pelo companheirismo, calma e alguns pouquíssimos puxões de orelha e por muitas outras qualidades que existem dentro daquele ser.

Aos meus professores Saulo Popov Zambiasi e Jusane Farina Lara, por me darem grande apoio quando precisei e por serem pessoas incríveis em suas profissões, dando-me orgulho por eu seguir este caminho.

Aos os meus pais Vilson e Delise por fazerem indiretamente, e diretamente, este trabalho comigo sendo tão bondosos quanto corretos, me permitindo, não só nestes últimos cinco anos, mas em toda minha vida, ter dois grandes exemplos de vida honesta, verdadeira e feliz, amo vocês.

Aos dois anjos que Deus pos em minha vida como irmã e companheira. Sahra e Lara, obrigado por existirem e por me darem o que eu mais precisava nas horas difíceis, amor e carinho.

A Deus por estar sempre ao meu lado, mesmo quando eu o esquecia. Iluminando-me e dando-me coragem para sempre seguir em frente.

SUMÁRIO

LISTA DE FIGURAS	x
LISTA DE TABELAS E QUADROS	xiv
LISTA DE SIGLAS E ABREVIATURAS	xv
1 INTRODUÇÃO.....	19
1.1 OBJETIVOS.....	20
1.1.1 Geral.....	20
1.1.2 Específicos	20
1.2 ORGANIZAÇÃO DO TRABALHO.....	20
2 TECNOLOGIA WEB	22
2.1 HIPERTEXTO, MULTIMÍDIA E HIPERMÍDIA	22
2.2 MODELOS E METODOLOGIAS DE DESENVOLVIMENTO HIPERMÍDIA ..	23
2.3 GERENCIAMENTO DE CONTEÚDOS	28
2.4 MODELOS DE DESENVOLVIMENTO.....	29
2.5 LINGUAGENS E PARADIGMAS	31
2.6 APLICAÇÕES BASEADAS NA WEB (WEBAPPS).....	33
3 ENGENHARIA WEB	34
3.1 ENGENHARIA.....	34
3.2 DEFINIÇÃO DA ENGENHARIA WEB (WEBE)	36
3.3 REQUISITOS PARA A QUALIDADE DE UMA WEBAPP	36
3.4 PROCESSO DE DESENVOLVIMENTO.....	38
3.5 ETAPAS DO PROCESSO	39

3.5.1	Formulação, Planejamento e Análise	39
3.5.2	Projeto	40
3.5.3	Implementação e Testes	45
3.5.4	Avaliação do Cliente	46
4	METODOLOGIA OOHDMD	47
4.1	O OOHDMD.....	47
4.2	LEVANTAMENTO DE REQUISITOS	52
4.3	MODELAGEM CONCEITUAL	54
4.3.1	Componentes do Esquema Conceitual.....	54
4.3.2	Esquema Conceitual	59
4.4	MODELAGEM NAVEGACIONAL	60
4.4.1	Esquema de Classes Navegacionais.....	60
4.4.2	Esquema de Contexto de Navegação	63
4.4.3	Esquema de Classes em Contexto e Cartões de Identificação.....	65
4.5	PROJETO DA INTERFACE ABSTRATA	67
4.5.1	Diagramas de Configuração.....	67
4.5.2	ADVcharts	68
4.6	IMPLEMENTAÇÃO	69
5	ESCOTISMO	71
5.1	DEFINIÇÕES E CONCEITOS.....	73
5.2	ESTRUTURA ORGANIZACIONAL NO ESCOTISMO	74
5.2.1	O Grupo Escoteiro (GE)	76
5.3	FICHAS E CERTIFICADOS.....	77

5.3.1	Fichas REME.....	77
5.3.2	Ficha 120 e 121.....	77
5.3.3	Certificados.....	79
5.4	GE XIMBANGUE.....	80
6	ENGENHARIA DE UMA WEBAPP PARA UM GRUPO ESCOTEIRO UTILIZANDO O MÉTODO OOHD.....	81
6.1	LEVANTAMENTO DE REQUISITOS	82
6.1.1	Identificação de Atores e Tarefas.....	84
6.1.2	Especificação de Cenários.....	85
6.1.3	Especificação dos Use Cases.....	88
6.1.4	Especificação dos UIDs	90
6.1.5	Validação dos Use Cases e UIDs	91
6.2	AS MODELAGENS	92
6.3	MODELAGEM CONCEITUAL	93
6.4	MODELAGEM NAVEGACIONAL	96
6.4.1	Esquema de Contextos	96
6.4.2	Esquema Navegacional	101
6.4.3	Cartões de Especificação.....	106
6.5	MODELAGEM DE INTERACE ABSTRATA	108
6.6	IMPLEMENTAÇÃO	111
6.6.1	Mapeamento para modelo Relacional	111
6.6.2	Desenvolvimento e Testes	114
6.7	AVALIAÇÃO E MANUTENÇÃO.....	117

7	CONSIDERAÇÕES FINAIS.....	118
7.1	RESUMO DAS CONTRIBUIÇÕES.....	118
7.2	TRABALHOS FUTUROS.....	119
8	REFERÊNCIAS.....	121
9	ANEXOS.....	125
9.1	RELAÇÃO DE CENÁRIOS.....	125
9.2	RELAÇÃO DE USE CASES.....	141
9.3	RELAÇÃO DE UIDS.....	145
9.4	RELAÇÃO DE CARTÕES DE VISÃO.....	154
9.4.1	Ator: DIRIGENTE	154
9.4.2	Ator: CHEFE.....	156
9.4.3	Ator: ESCOTEIRO.....	158
9.5	RELAÇÃO DE CARTÕES DE ESPECIFICAÇÃO.....	160
9.5.1	Cartões de Contexto	160
9.5.2	Cartões de Estrutura de Acesso	164
9.6	RELAÇÃO DE ADVS	167
9.7	RELAÇÃO DE DIAGRAMAS DE CONFIGURAÇÃO.....	171

LISTA DE FIGURAS

<i>Figura 1: Modelo MVC.....</i>	30
<i>Figura 2: Ambiente da Linguagem PHP</i>	32
<i>Figura 3: Ilustração baseada no modelo de processo da WebE.</i>	34
<i>Figura 4: Camadas da Engenharia de Software.....</i>	35
<i>Figura 5: Árvore de Olsina para representação de Requisitos de qualidade da WebApp.</i>	37
<i>Figura 6: Estruturas Lineares.....</i>	42
<i>Figura 7: Estrutura em Malha.</i>	42
<i>Figura 8: Estrutura Hierárquica.....</i>	43
<i>Figura 9: Estrutura Interligada ou Pura teia.....</i>	43
<i>Figura 10: Ciclo de desenvolvimento usando OOHDM.</i>	51
<i>Figura 11: Ciclo de desenvolvimento usando OOHDM.</i>	52
<i>Figura 12: Representação de uma classe.....</i>	54
<i>Figura 13: Exemplo de uma classe do esquema conceitual.....</i>	55
<i>Figura 14: Exemplo de generalização/especialização.</i>	56
<i>Figura 15: Exemplo de Relacionamentos.....</i>	57
<i>Figura 16: Relacionamento 1-N.....</i>	58
<i>Figura 17: Esquema Conceitual de um Domínio Acadêmico</i>	59
<i>Figura 18: Diagrama de objetos.....</i>	62
<i>Figura 19: Exemplo de um Diagrama Navegacional.....</i>	62
<i>Figura 20: Esquema de contexto de navegação.....</i>	64
<i>Figura 21: Classes em Contexto.</i>	65
<i>Figura 22: Exemplos de Cartões de Identificação.....</i>	66

<i>Figura 23: Diagrama de Configuração.....</i>	68
<i>Figura 24: ADVchart Laboratório.....</i>	69
<i>Figura 25: Emblema da UEB.....</i>	74
<i>Figura 26: Símbolo do Lobismo.....</i>	74
<i>Figura 27: Símbolo do Ramo Escoteiro</i>	75
<i>Figura 28: Símbolo do Ramo Sênior.....</i>	75
<i>Figura 29: Símbolo do Ramo Escoteiro.</i>	75
<i>Figura 29: Modelo de Ficha 120.</i>	78
<i>Figura 31: Logotipo do GE Ximbangue.</i>	80
<i>Figura 32: Estrutura de Desenvolvimento da WebApp.....</i>	81
<i>Figura 33: UID 02 – Cadastrar Seção</i>	90
<i>Figura 34: Modelagens OOHDM</i>	92
<i>Figura 35: Modelo Conceitual da WebApp do GE Ximbangue.....</i>	95
<i>Figura 36: Contexto Navegacional Ator Dirigente</i>	98
<i>Figura 37: Contexto Navegacional Ator Chefe.....</i>	99
<i>Figura 38: Contexto Navegacional Ator Escoteiro</i>	100
<i>Figura 39: Diagrama Navegacional Ator Dirigente</i>	103
<i>Figura 40: Diagrama Navegacional Ator Chefe.....</i>	104
<i>Figura 41: Diagrama Navegacional Ator Escoteiro</i>	105
<i>Figura 42: ADV Principal da WebApp.....</i>	108
<i>Figura 43: ADV Conteúdo da WebApp.</i>	109
<i>Figura 44: Diagrama de Configuração.....</i>	109
<i>Figura 45: ADVCharts de Autenticação de usuário.....</i>	110
<i>Figura 46: ADVCharts de Controle de menus.....</i>	110

<i>Figura 47: Diagrama de Modelagem-Relacionamento da WebApp.</i>	113
<i>Figura 48: Screenshot Visualizar Grupo Escoteiro.</i>	115
<i>Figura 49: Screenshot Editar Grupo Escoteiro.</i>	115
<i>Figura 50: Screenshot Visualizar Seção.</i>	116
<i>Figura 51: Screenshot Cadastrar Atividade.</i>	116
<i>Figura 52: UID 01 Alterar informações institucionais do GE.</i>	145
<i>Figura 53: UID 02 Cadastrar Seção.</i>	145
<i>Figura 54: UID 03 Cadastrar Membro.</i>	146
<i>Figura 55: UID Ver relação de escoteiros por seção.</i>	147
<i>Figura 56: UID 05 Ver ficha 120.</i>	148
<i>Figura 57: UID 06 Ver membros por seção.</i>	149
<i>Figura 58: UID 07 Transferir um membro para uma seção.</i>	150
<i>Figura 59: IUD 08 Registrar datas significativas.</i>	151
<i>Figura 60: UID 09 Registrar Atividades.</i>	152
<i>Figura 61: UID 10 Ver detalhes de um membro.</i>	152
<i>Figura 62: UID 11 Alterar dados pessoais.</i>	153
<i>Figura 63: ADV de Estrutura da WebApp.</i>	167
<i>Figura 64: ADVs de Conteúdo da WebApp.</i>	167
<i>Figura 65: ADVs atômicos da WebApp.</i>	168
<i>Figura 66: ADV de Autenticação de Usuário.</i>	168
<i>Figura 67: ADVs de Visualização e Edição de GE.</i>	168
<i>Figura 68: ADVs de Visualização, Edição e Listagem de Seções.</i>	169
<i>Figura 69: ADVs de Visualização e Edição de Seções.</i>	169
<i>Figura 70: ADVs de Visualização e Edição de Membros.</i>	170

<i>Figura 71: ADVs de Criação, Edição e Listagem de Membros.....</i>	170
<i>Figura 72: Diagrama de configuração do ADO GE.....</i>	171
<i>Figura 73: Diagrama de configuração do ADO Formulário de GE.....</i>	171
<i>Figura 74: Diagrama de configuração do ADO Seção.....</i>	172
<i>Figura 75: Diagrama de configuração do ADO Atividades.....</i>	173
<i>Figura 76: Diagrama de configuração do ADO Formulário de Atividades.....</i>	174
<i>Figura 77: Diagrama de configuração do ADO Formulário Membro.....</i>	175
<i>Figura 78: Diagrama de configuração do ADO Escoteiro.....</i>	175
<i>Figura 79: Diagrama de configuração do ADO Chefe.....</i>	176
<i>Figura 80: Diagrama de configuração do ADO Dirigente.....</i>	176

LISTA DE TABELAS E QUADROS

<i>Tabela 1: Comparação de Modelos e Metodologias.....</i>	27
<i>Tabela 2: Tabela de Características herdadas e criadas no OOHDM</i>	49
<i>Tabela 3: Exemplos de cenários da WebApp para um GE</i>	87
<i>Tabela 4: Exemplo de um use cases da WebApp para um GE.....</i>	89
<i>Tabela 5: Exemplos de Cartões de Visão</i>	101
<i>Tabela 6: Exemplo de Cartão de Contexto.....</i>	106
<i>Tabela 7: Exemplo de Cartão de Estrutura de Acesso</i>	107
<i>Quadro 1: Esboço da metodologia OOHDM.....</i>	50
<i>Quadro 2 : Graus de cardinalidade em um relacionamento.....</i>	57

LISTA DE SIGLAS E ABREVIATURAS

ADO	Object Data View
ADV	Abstract Data View
ARPA	Advanced Research and Projects Agency
ASP	Active Server Page
B-P	Baden-Powell
CSS	Cascading Style Sheet
ERMIA	Entity-Relationship Modeling of Information
GE	Grupo Escoteiro
HDM	Hypermedia Design Model
HTML	Hypertext Markup Language
JSP	Java Server Page
MACPRO	Método de Atualização e Criação Permanente do Programa de Jovens
MVC	Model, View and Controller
OO	Orientado a Objetos
OOHDM	Object Oriented Hypermedia Design Model
PHP	PHP Hypertext Preprocessor
RMM	Relationship Management Methodology
SGBD	Sistema gerenciador de Banco de Dados
SQL	Structured Query Language
UEB	União dos Escoteiros do Brasil
UID	User Interator Diagram
UML	Unified Modeling Language

W3C	World Wide Web Consortium
WebML	The Web Modeling Language
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

RESUMO

Este trabalho tem por objetivo: contextualizar o ambiente web, suas tecnologias, paradigmas, ambientes e metodologias de desenvolvimento; fazer um estudo sobre a engenharia aplicada a web, seus métodos, princípios e processos; levantar subsídios teóricos para realizar a aplicação desta engenharia através do método OOHDM para o desenvolvimento de uma WebApp para um Grupo Escoteiro; e um estudo sobre o relacionamento entre os processos formais da engenharia web e as soluções propostas pelo método OOHDM através deste caso de uso real.

ABSTRACT

The objective of this final paper is: contextualize a web ambient, its technologies, paradigms, development ambient and methodologies; to make a study about the web engineering, its methods, principles and process; to get theory bases to make the application of it engineering by OOHDM method for the development of a WebApp to a Scout Group; and a study about the relationship between the formal process of web engineering and the solutions proposed by the OOHDM.

1 INTRODUÇÃO

Tanto para Ginige (2002), quanto para Pressman (2002), as aplicações baseadas na Web, ou simplesmente WebApps, necessitam de uma engenharia, como qualquer outro produto que se queira desenvolver com qualidade. Porém, segundo estes autores, muitos dos atuais desenvolvedores de aplicações web não aceitam esta idéia de desenvolver seus produtos de forma tradicional (com uma engenharia) e encaram seus trabalhos como obra de artes ou simplesmente, algo que deva funcionar para eles, e não seus usuários.

Fatos como estes acarretam: 84% em aplicações que não atendem as necessidades do mercado, 53% não cobrem suas funcionalidades requeridas, 63% excedem seus orçamentos, sendo que 79% destes contratemplos resultam de problemas no projeto (GINIGE, 2002).

Porém ao se aplicar uma engenharia sobre o desenvolvimento de WebApps, o desenvolvedor encontra outra grande incógnita pois, segundo Medeiros e Schwabe (1998), muitas são as linguagens declarativas para especificações de projeto em aplicações hipermídia. Dentre elas pode-se citar a HDM (*Hypermedia Design Model*) e a WebML. Porém, estas possuem certos aspectos que, por sua vez, podem dificultar o projeto, exigindo, em algum momento, a realização de atividades que poderiam ser postergadas. O OOHDMD (*Object-Oriented Hypermedia Design Model*) vem se apresentando muito satisfatório na comunidade hipermídia, sendo bem aceito em diversos países. Uma das características deste modelo é sua capacidade de separar claramente as fase da engenharia e permitir que estas sejam trabalhadas em paralelo, até por equipes diferentes.

Este trabalho surgiu da necessidade de aplicar conceitos de engenharia e modelagem web, em seus respectivos passos, perante todo ciclo necessário, para desenvolver um protótipo de uma WebApp modelada pelo OOHDMD. Explanando de forma simples e direta cada atividade realizada. Garantindo que a aplicação se mantenha em um escopo gerenciável e seguro demonstrando os benefícios que uma modelagem, pode proporcionar a um projeto para web, em suas necessidades básicas conforme identifica Pressman (2002): imediatismo, segurança e estética.

Para uma comparação satisfatória e exemplificada da utilização do método OOHDMD no processo de desenvolvimento da engenharia web, fez-se necessário o estudo de problemas de um caso real onde a aplicação de uma solução on-line fosse o mais indicado. Neste contexto optou-se pelo desenvolvimento de uma solução para os problemas enfrentados pela maior fraternidade mundial chamada Escotismo. Tanto pelo conhecimento do autor deste

trabalho sobre esse assunto, como pelas inúmeras necessidades enfrentadas diariamente pelo movimento escoteiro.

Movimento este que apresenta aos jovens de 07 aos 21 anos um conjunto de opções de crescimentos tanto mental, físico, espiritual, afetivo e de caráter que requerem a sua escolha e sua dedicação, sendo que a proposta mais relevante é a incorporada na Lei Escoteira, em que se reúnem os principais valores de seu projeto educativo, dever para com Deus, com sua pátria e consigo mesmo (P.O.R., 2001).

1.1 OBJETIVOS

1.1.1 Geral

Prototipar uma WebApp, dentro dos passos indicados pela engenharia web, para o gerenciamento de um grupo escoteiro utilizando, como ferramenta de modelagem, o método OOHDm.

1.1.2 Específicos

Com este trabalho pretende-se alcançar os seguintes objetivos específicos:

- estudo dos fundamentos e procedimentos da engenharia web;
- estudo dos passos e estrutura do método OOHDm;
- reconhecimento dos resultados que são propostos pelo método OOHDm através de sua utilização prática;
- estudo e reconhecimento da estrutura organizacional e funcional de um grupo escoteiro.

1.2 ORGANIZAÇÃO DO TRABALHO

Esta monografia é composta por nove capítulos apresentados de forma evolutiva.

No primeiro capítulo é apresentada uma introdução sobre a que o conteúdo deste trabalho se dispõe a apresentar.

No segundo capítulo é realizada uma contextualização do universo web, suas tecnologias e seu funcionamento. Também é neste capítulo que se encontra uma visão geral dos métodos e metodologias pertinentes ao desenvolvimento de aplicações hipermídia.

No terceiro capítulo é apresentada a engenharia web, seus conceitos, processos e propósitos.

No quarto capítulo é dada a introdução ao método OOHDM, sua origem e seu formato, também justificando a utilização desta metodologia para este trabalho.

No quinto capítulo é realizada uma contextualização sobre o foco do caso de uso do trabalho: a estrutura organizacional e funcional do Escotismo e de um Grupo Escoteiro.

O sexto capítulo apresenta de forma direta os problemas abordados no quinto capítulo e qual deles será implementado com a engenharia web e o modelo OOHDM, bem como, a forma em que este será resolvido.

O sétimo e oitavo capítulos apresentarão as considerações alcançadas com o estudo deste trabalho, bem como, as fontes bibliográficas utilizadas para o desenvolvimento deste.

O nono capítulo apresentará a relação dos principais documentos gerados pelo OOHDM para o desenvolvimento do estudo de caso.

2 TECNOLOGIA WEB

Neste capítulo é realizada uma contextualização do universo web, suas tecnologias, breve histórico e seu funcionamento. Também neste capítulo será abordada uma visão geral dos métodos e metodologias pertinentes ao desenvolvimento de aplicações hipermídia.

A Web é uma rede de comunicação lógica que abrange várias partes do mundo através da estrutura física denominada Internet (MOLINARI, 2004).

Esta rede surgiu em 1969 na guerra fria com o objetivo de armazenar e compartilhar informações entre algumas cidades específicas dos Estados Unidos. Foi desenvolvida primeiramente pela ARPA (Advanced Research and Projects Agency) e teve seu primeiro nome como ARPANET.

A tecnologia da Internet sobreviveu a guerra e teve sua inclusão comercial e pública somente em 1992 e chegou no Brasil em 1995 (ALVES, 2004).

2.1 HIPERTEXTO, MULTIMÍDIA E HIPERMÍDIA

Segundo Bulterman (1994), citado por Costagliola (2002), e Hennrichs (2005), o hipertexto, em termos de informática, é modelado como uma rede de documentos, formado por várias páginas independentes, que são conectadas através de links. Cada página normalmente representa um contexto e os links são responsáveis pela coerente ligação entre estes contextos, permitindo uma “navegação de um texto mais lógica (não linear), fazendo com que o usuário crie seu próprio ritmo, nível e estilo, adequando às suas características e interesses” (HENNRICHS, 2005).

Uma apresentação multimídia é caracterizada por uma combinação de múltiplas mídias, como textos, imagens, sons e vídeos. A multimídia diferencia-se das mídias tradicionais, como as transmitidas via televisão publica, pelo fato de proporcionar a interação com quem esta a assistindo.

Hipermídia é a combinação entre o hipertexto e a multimídia, tendo para cada apresentação de um contexto do hipertexto uma interatividade multimídia. Links podem ser responsáveis por, além de conectar a outros contextos relacionados ao contexto que está sendo apresentado, interagir com os objetos apresentados na multimídia, como sons, imagens e vídeos (COSTAGLIOLA ,2002).

2.2 MODELOS E METODOLOGIAS DE DESENVOLVIMENTO HIPERMÍDIA

Em uma pesquisa realizada pelo professor Gennaro Costagliola, junto com Filomena Ferrucci e Rita Francese, em 2002, *chamada Web Enginerring: Models and Methodologies for the Design of Hypermedia Applications*¹ (Engenharia Web: Modelos e Metodologias para Projetos de Aplicações Hipermídia) foram feitos comparativos dos principais modelos e metodologias de design/projetos² hipermídia.

Modelos de desenvolvimento hipermídia podem ser divididos em dois grupos, modelos representacionais e modelos de projeto.

Conforme Halasz e Paolini (1994), citados na pesquisa do professor Costagliola, os modelos representacionais são responsáveis pela organização nas entrelinhas de um hipertexto, auxiliando o desenvolvedor em uma melhor compreensão e manutenção das informações contidas no conteúdo dos hipertextos, permitindo o desenvolvimento de aplicações mais portáveis e interoperáveis.

Os modelos de projeto, descrito por Pressman (2000) citado por Costagliola (2002), permite um conhecimento mais profundo sobre como os objetos de uma aplicação surgem e quais suas reais tarefas. Como o modelo relacional auxilia o entendimento e comunicação entre os desenvolvedores de uma estrutura de dados, na abordagem hipermídia essa documentação (links, estrutura, navegação, entre outros) é fornecida pelos modelos de projeto.

Na pesquisa do professor Costagliola foram estudados três modelos de desenvolvimento hipermídia dispostos abaixo.

- **Trellis Hypermedia Reference Model** (Modelo de Referência Hipermídia): baseado na representação *Petri Net* para estrutura de documentos, este modelo pode ser considerado um modelo comportamental de hipertexto.

¹ Acessado em Abril de 2005 no site: <ftp://cs.pitt.edu/chang/handbook/58b.pdf>

² Design, palavra em inglês que significa Projeto

- **HDM – Hipermídia Design Model** (Modelo de Projetos Hipermídia): baseado no modelo entidade-relacionamento. O HDM permite ao projetista identificar os componentes atômicos e os critérios da montagem da estrutura do projeto.
- **UML³ extensions to Hypermedia Applications** (UML Extensões para Hipermídia Aplicações): a UML pode ser considerada um padrão no desenvolvimento de softwares orientados-objeto. Ela não é uma metodologia, mas sim uma linguagem de modelagem. Além de servir de base para a metodologia OOHDMD, que será visto melhor adiante, iniciou-se a adaptação de algumas extensões no modelo UML, que tem por objetivo englobar o contexto da hipermídia dentro da modelagem.

Conforme a dissertação de Roque (1998), dentre autores como Avison e Fitzgerald (1997), Fitzgerald (1997) e Yourdon (1995), o entendimento de metodologia se dá a um conjunto pré-definido de atividades – postadas como documentos, procedimentos e técnicas, e algumas vezes com o auxílio de ferramentas – que devem ser executados corretamente com o objetivo final de desenvolver um sistema de informação de alta qualidade.

Segundo Costagliola (2002), não há a metodologia perfeita, mas sim, uma metodologia mais apropriada para cada produto a ser desenvolvido. Lembrando ainda que muitas metodologias podem ser praticadas em diferentes fases de um mesmo projeto, devido ao fato que cada uma possuir uma melhor solução para esta fase.

Na pesquisa de Costagliola (2002) foram comparadas quatro diferentes metodologias: a RMM, o OOHDMD, os Design Patterns e o Ermia, conforme descritos abaixo.

O **RMM⁴** auxilia na fase de projeto das aplicações hipermídia que possuem a necessidade de várias informações de diferentes fontes no mesmo layout de apresentação. Baseia-se nas primitivas RMDM (Relationship Management Data Model – Modelo de gerenciamento de relacionamentos de dados) e pode ser dividido entre primitivas ER

³ UML: The Unified Modeling Language – Linguagem de Modelação Unificada.

⁴ RMM: The Relationship Management Methodology – Metodologia de gerenciamento de relacionamentos.

(entidade-relacionamento), primitivas de domínio de camadas (m-Slice domain primitives) e primitivas de acesso (ISAKOWITZ, 1996).

O **OOHDM**⁵, baseado no modelo HDM, visto anteriormente, permite o desenvolvimento de aplicações hipermídia por um ciclo de vida de desenvolvimento prototipado incremental. Esta metodologia é dividida em cinco fases: levantamento de requisitos, modelagem conceitual, modelagem navegacional, projeto de interface abstrata e implementação.

Segundo Costagliola (2002):

The OOHDM methodology improves maintainability and reusability, thanks to both the separation of the design phases and the abstraction capabilities which are characteristics of the object-oriented design, isto é, a metodologia OOHDM melhora a manutenibilidade e a reusabilidade, permitidos pela separação das fases do projeto e pelas potencialidades de abstração, que são características dos projetos orientados a objeto.

Design Patterns⁶, segundo Bolchini (2000), podem ser formalizados como soluções de projetos decorrentes de padrões de problemas já predefinidos. Permitindo que os conhecimentos de projetistas experientes e os recursos hipermídia possam ser compartilhados de forma simples e direta, auxiliando tanto novos desenvolvedores como desenvolvedores mais avançados.

ERMIA⁷, para Green e Benyon (1996), é um método de design de alto nível. Diretamente baseado no modelo entidade-relacionamento, ele proporciona ao projetista uma visão tanto abstrata, como ocorre no início do projeto, até uma visão bem definida da estrutura da aplicação. Possui suporte para análise de custos. O ERMIA possui cinco propostas de estruturas internas: pilha (pile), corrente (chain), lista classificada (sorted list), *hashed* e *unsearchable*.

⁵ OOHDM: Object-Oriented Hypermedia Design Model – Modelagem de projeto hipermídia orientado-objeto.

⁶ Design Patterns: Padrões de Projeto

⁷ ERMIA: Entity-Relationship Modeling of Information Artifacts – Modelagem entidade-relacionamento para artefatos de informação.

Essas quatro metodologias, mencionadas acima, foram analisadas nos seguintes quesitos:

- **metodologia:** uma consideração geral em seus métodos e seqüências de atividades;
- **contemplação:** a robustez, no que diz respeito a melhor obtenção e armazenamento das informações, da metodologia perante todo ciclo de desenvolvimento da aplicação;
- **modelo:** a ligação da metodologia com algum modelo de desenvolvimento.
- **ferramentas CASE:** a existência e suporte de alguma ferramenta de suporte para essa metodologia;
- **aspectos cognitivos:** a definição da seqüência de atividades para um melhor entendimento do problema a ser resolvido e a lógica da solução;
- **métricas de custo:** análise dos custos do projeto;
- **sincronização:** a facilidade do projetista em descrever os relacionamentos temporais entre as diferentes mídias.

O estudo comparativo de Costagliola (2002) demonstra que não há uma metodologia perfeita e completa, mas sim, diferentes tipos de metodologias para diferentes tipos de aplicações e obteve como resultado as seguintes conclusões:

- O HDM baseia-se em uma extensão do modelo entidade-relacionamento de dados, permitindo uma descrição global da estrutura de aplicações multimídia.
- O OOHDM, por ter sua origem no modelo HDM, e ser uma estrutura baseada na orientação a objetos, se torna uma metodologia de desenvolvimento de grandes aplicações hipermídia através de um ciclo de desenvolvimento incremental e baseado em prototipação.
- O RMM permite o desenvolvimento de *web sites* com foco em aspectos navegacionais e com integração a banco de dados de grande porte.

- O modelo Trellis permite especificações da estrutura dos hipertextos e aspectos de sincronização entre eles.
- O ERMIA é um método de projetos de alto nível e suporta análises de custo.
- As extenções UML permitem a integração entre a UML e características de modelagens navegacionais.

Na tabela abaixo (Tabela 1) encontra-se o comparativo entre os modelos e metodologias abordados na pesquisa de Costagliola (2002).

Tabela 1: Comparação de Modelos e Metodologias

	RMM	HDM	OOHDM	ERMIA	TRELLIS	UML ext.
Metodologia	X		X			
Contemplação			X			X
Modelo	X	X	X	X	X	X
Perspectiva/Contexto		X	X			
Ferramentas CASE	X				X	
Aspectos cognitivos			X	X		
Métricas de custo				X		
Sincronização			X		X	X

Fonte: Costagliola (2002).

A tabela de comparação de modelos e metodologias (Tabela 1) apresenta a superioridade da metodologia OOHDM sobre as outras metodologias comparadas, permitindo visualizar o grande grau de integração desta metodologia com o processo de desenvolvimento real de aplicações multimídia.

2.3 GERENCIAMENTO DE CONTEÚDOS

Para Content Management (2004) “Content is in essence, any type or 'unit' of digital information. It can be text, images, graphics, video, sound, documents, records etc - or in other words - anything that is likely to be managed in an electronic format”, ou seja, conteúdo é qualquer tipo de informação digital, seja ela textos, imagens, sons, documentos, gravações ou outros, sendo assim, qualquer coisa que possa ser gerenciada em formato digital.

Segundo a AppliedTheory, apud Moratelli (2002): “Gerenciamento do conteúdo é o controle – administração, gerenciamento de fluxo, acesso ao conteúdo e segurança das informações de uma organização.”

O gerenciamento de conteúdo para o ambiente Web necessita de ferramentas rápidas e de fácil implementação, tendo em vista a constante evolução dos aplicativos desenvolvidos nesta área. Também exige programas seguros por se tratar de uma aplicação que disponibiliza conteúdo para diversos locais e por diversos tipos de usuários.

Segundo a INFOIMAGE

Gestão de Conteúdos é uma aplicação focada na montagem de componentes de informação a alta velocidade e na publicação do resultado, tipicamente na Internet. Tem como principais predicados a sua capacidade de operação dinâmica e rápida e a capacidade de injectar os requisitos individuais dos leitores destinatários em conteúdos e formatos diferentes. Embora possa parecer que gere documentos, de facto não o faz. De facto o que gere é um stock de componentes, alguns dos quais serão até documentos enquanto outros são uma variedade de elementos multimedia como videos, graficos, audio. Separa o conteúdo (informação ou dados) das regras de negócio (para o que é que os dados servem) e da apresentação (como é que a informação deve ser acedida e vista pela audiência) (2001).

Segundo Fagundes⁸ (2004), aplicações web que necessitam de gerenciamento de conteúdo sempre utilizarão SGBD's⁹. Sendo que há vários bancos de dados no mercado, o mais utilizado hoje em aplicações web é o MySQL.

⁸ Acessado em 15 de Abril de 2005 no endereço web:
<http://www.vivaolinux.com.br/artigos/impressora.php?codigo=815>

⁹ SGBD: Sistema gerenciador de banco de dados.

De acordo com Choi et al (2001), citado por Silva et al (2003), o SGBD MySQL apresenta características muito apreciáveis em aplicações web, como:

- ser uma aplicação open source¹⁰, ou seja, trata-se de uma aplicação que pode ser usada e alterada sem investimentos financeiros.
- possuir suporte a linguagem SQL (Structured Query Language), que é padrão em banco de dados relacionais.
- apresentar alta performance e confiabilidade, que são requisitos básicos em aplicações web.
- excelente portabilidade, permitindo a utilização da mesma estrutura do banco tanto para ambientes Windows com UNIX e Linux.

2.4 MODELOS DE DESENVOLVIMENTO

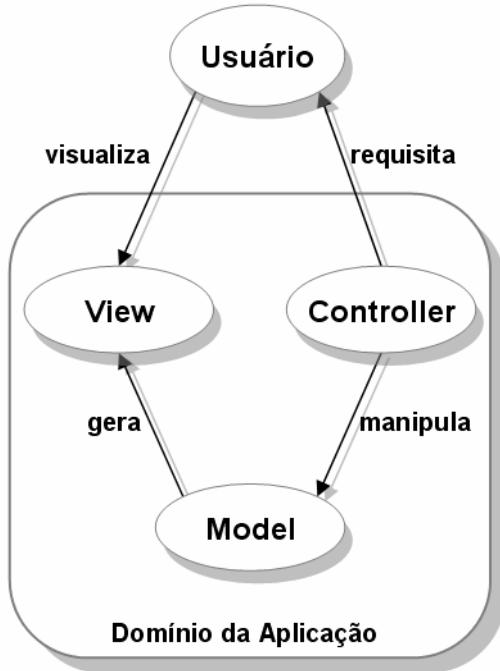
Para atender as expectativas de uma aplicação onde: as operações devem ser dinâmicas e rápidas; as informações possuem fontes heterogêneas, ou seja, vários tipos de editores; e as informações devem possuir uma apresentação em formatos padronizados por diferentes perfis de usuários, existe várias técnicas. Um modelo bastante conhecido na comunidade de desenvolvedores web é o padrão MVC utilizando STRUTS, um *framework* de código aberto para o desenvolvimento de aplicações web em Java (WIZARD STRUTS¹¹,2005).

O modelo MVC consiste na divisão da aplicação em três camadas distintas: Model, View e Controller. A camada Model, ou Modelo, é responsável pela lógica da aplicação, bem como, pela comunicação da aplicação com o banco de dados. A camada View, ou Visualização, é responsável pela interface de apresentação do conteúdo montado pela classe Model. A camada Controller, ou Controladora, controla as requisições do usuário perante o que a camada View permite que o usuário faça. Fazendo com que sejam executadas novas

¹⁰ Open source: código livre.

¹¹ Acessado em 10 de Abril de 2005 no endereço web:
[<http://www.qualiti.com.br/arquivos/coder/manualWizardStruts.PDF>](http://www.qualiti.com.br/arquivos/coder/manualWizardStruts.PDF)

instruções na camada Model. A figura a seguir ilustra a estrutura do modelo MVC conforme a fonte Wizard STRUTS (2005).



Fonte: Wizard STRUTS (2005).

Figura 1: Modelo MVC.

O modelo MVC apresentado na figura 1 permite as seguintes vantagens no desenvolvimento de aplicações:

- as interfaces se tornam independentes do processo interno da aplicação, no que diz respeito a recuperação das informações nos repositórios de dados e a lógica de manipulação desses dados;
- da mesma forma, a lógica da aplicação se torna independente da forma, *layout*, de apresentação dos dados, permitindo uma maior liberdade tanto por parte dos desenvolvedores como por parte dos *designers*¹² do projeto.

¹² Designers: equipe responsável pela apresentação gráfica das aplicações, formatação e disposição de ferramentas e conteúdos.

2.5 LINGUAGENS E PARADIGMAS

Um fator de suma importância para se levar em consideração do desenvolvimento de aplicações é a linguagem que será utilizada pra desenvolver os programas do sistema. Hoje há diversas linguagens disponíveis para o desenvolvimento de aplicações hipermídia, estas linguagens pode ser dividida em dois grupos: cliente e servidor.

Conallen (2003) afirma que há três elementos fundamentais de um sistema Web: o **navegador cliente**, a **rede** e um **servidor Web**. Elementos que permitem que as páginas Web sejam solicitadas pelo navegador cliente ao servidor Web através da rede.

Na Internet o padrão de desenvolvimento de páginas desde seu surgimento, até os dias atuais, foi o HTML (Hypertext Markup Language), desenvolvido pela W3C¹³ (2004). Trata-se de uma linguagem de formatação de documentos, permitindo ao programador inserir textos e imagens intercalados em tabelas e outras formatações de visualização e também inserir ligações (links) destas páginas a outras.

Hoje, em sua versão 4.0.1, o HTML está sendo reestruturado por seus criadores para uma nova linguagem, o XHTML, que segundo a W3C será a linguagem de conteúdo do futuro da Web.

O HTML possui indiscutível utilidade no desenvolvimento de aplicações web, porém apresenta restrições óbvias perante o desenvolvimento de sistemas:

- é uma linguagem estática, não permitindo a fácil e prática manutenção de seu conteúdo.
- é uma linguagem interpretada no cliente, ou seja, permite a qualquer usuário que visualize a pagina, visualizar também seu código fonte.

Para estes inconvenientes foram desenvolvidas linguagens de servidor para o desenvolvimento de aplicações mais dinâmicas e restritas, a respeito de conteúdo e lógica da aplicação, sendo algumas destas o ASP .NET, o Java e o PHP.

¹³ W3C: World Wide Web Consortium. Acessado em 17 de Abril de 2005 no site: <www.w3c.org>

Segundo a WebEstilo¹⁴ (2005), o ASP (Active Server Pages) é uma linguagem desenvolvida pela Microsoft para criar páginas de conteúdos dinâmicos utilizando scripts executados no servidor. Castagneto et al (2001) explica que uma página ASP é composta por comandos HTML intercalada com scripts escritos em ASP, e ao ser localizada pelo servidor, por uma requisição por parte de um cliente, é executada, tendo seus scripts interpretados, gerando ao final um arquivo somente em HTML, o qual é enviado ao cliente.

Segundo Castagneto et al (2001), a linguagem servidor Java para ambientes web, divide-se em duas tecnologias:

- Sevlets Java: é um programa no servidor que se comunica com o cliente através de uma conexão http;
- Java Server Pages (JSP): semelhante a ASP, uma página JSP é composta por HTML e scripts JSP, e quando requisitada por um cliente, é interpretada formando somente um arquivo HTML.

O PHP (PHP Hypertext Processor), conforme apresentado pela WebEstilo (2005) e pela PHP.net¹⁵ (2005), similar a linguagem ASP e Java atende da mesma forma as requisições de um cliente web, conforme a figura abaixo:



Fonte: WebEstilo (2005).

Figura 2: Ambiente da Linguagem PHP

O PHP em sua versão 5.0.2, ou simplesmente, PHP5, está com sua estrutura de orientação a objeto bem definida, com sua comunicação ao banco de dados MySQL4.x

¹⁴ Acessada na página: <www.webestilo.com.br/aspnet>. Acessada em Julho de 2005.

¹⁵ Acessada na página oficial do PHP: <www.php.net>. Acessada em Julho de 2005.

reestruturada, aproveitando melhor os recursos deste, e um maior suporte ao XML . Conforme foi apresentado em uma reportagem chamada “O PHP nasce de novo” sobre a mesma, na edição do mês de setembro da Infoexame (2004). Esta reportagem ainda traz dados estatísticos afirmando que o PHP roda em 16,9 milhões de sites, sendo este resultado 31,9% de um domínio de 53,1 milhões de sites existentes na segunda metade do ano de 2004, segundo dados da Netcraft.

2.6 APLICAÇÕES BASEADAS NA WEB (WEBAPPS)

Segundo Kappel (2004) e Pressman (2002) webapp são sistemas de aplicações, ou simplesmente softwares, baseados em tecnologias e padrões da Web, os quais são relacionados e organizados em complexas matrizes de conteúdo e funcionalidades, tendo em vista um grande número de usuários e um acesso a uma ou mais bases de dados, através de uma interface, o navegador *web*, também conhecido como *browser*.

Podem ser destacadas as seguintes características em um WebApp:

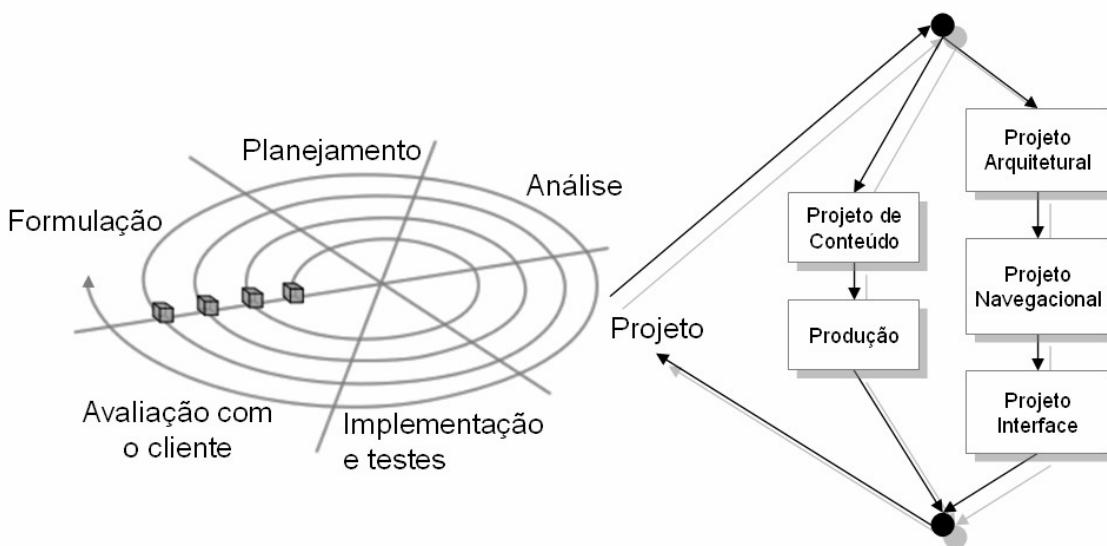
Imediatismo: A alta evolução da tecnologia e dos conteúdos pertinentes a uma WebApp exigem uma extrema velocidade de produção. Para isto, há necessidade de modelos que comportem esta velocidade.

Segurança: Por se tratar de uma aplicação Web, o controle sobre sua população de usuários é limitado a alguns requisitos de segurança formais (login e senhas, sessões, protocolos https ou ftps, restrição de IP's e outros), que não garantem total segurança de conteúdo, exigindo assim, uma infraestrutura de segurança bem implementada que dará um maior suporte à aplicação (PRESSMAN, 2002).

Estética: “A usabilidade governa a Web. Mais diretamente, se o cliente não encontrar o produto, ele não o comprará (...) A Web é o ambiente no qual o poder do cliente se manifesta no mais alto grau (...) todos concorrentes do mundo estão a um simples clique do mouse” Nielsen (2000) deixa bem claro a importância da primeira impressão das aplicações on-line, principalmente tratando-se de mundo Web, onde a informação está a distância invisível de alguns ‘cliques do mouse’. A obrigatoriedade de um sistema bem definido e funcionalmente correto é apenas parte de todo cuidado que deve-se ter sobre a interface de uma WebApp.

3 ENGENHARIA WEB

A engenharia web, também apresentada por inúmeros autores como WebE, demonstra-se como um processo cíclico e incremental de desenvolvimento e interação de aplicativos online. Dentre as bibliografias estudadas para esta pesquisa sobre engenharia web, tomou-se destaque, a seqüência de atividades apresentadas por Pressman (2002), que apresenta uma proposta de desenvolvimento, em uma seqüência direta e bem apresentável, em aplicações para Internet (Figura 3). Conforme apresentado na figura abaixo:



Fonte: Pressman (2002, p. 757).

Figura 3: Ilustração baseada no modelo de processo da WebE.

Mais detalhes sobre o processo de desenvolvimento da WebE, bem como, uma visão mais detalhada da figura acima é abordada no sub-capítulo “3.4 PROCESSO DE DESENVOLVIMENTO” deste capítulo.

3.1 ENGENHARIA

A engenharia consiste na aplicação de princípios empíricos, matemáticos e científicos, como facilitadores na busca da melhor forma de projetar, desenvolver e manter estruturas, dispositivos ou processos de forma a atender as necessidades humanas (The American Heritage Dictionary of English Language, apud GINIGE, 2002; AURÉLIO, 1999).

A engenharia possui diversas camadas de aplicação, nas quais cada uma especializa-se em um determinado meio. A engenharia de software pode entrar na forma de uma dessas camadas, como uma abordagem disciplinada para alocação de tarefas e responsabilidades, no processo de desenvolvimento de um sistema, auxiliando no desenvolvimento de produtos de alta qualidade, satisfazendo as necessidades dos usuários finais, em um cronograma e orçamento pré-estabelecidos (ÁLVARES,2000).

A engenharia de software compõe-se de três elementos fundamentais: métodos, ferramentas e procedimentos. Esses três elementos têm como objetivo auxiliar a visualização e tomada de decisões nos processos de desenvolvimento de uma aplicação, sempre focando a qualidade final do produto conforme a figura abaixo (PRESSMAN,2002).



Fonte: Pressman (2002, p. 19).

Figura 4: Camadas da Engenharia de Software.

Os métodos são responsáveis por estabelecer modelos básicos das atividades de cada área de uma aplicação: a base na elaboração das estimativas do projeto, da análise de requisitos, cronograma, codificação, teste, manutenção e orçamento.

As ferramentas são tecnologias que auxiliam a manipulação dos métodos. Quando essas ferramentas englobam e integram todos os métodos a serem utilizados para o desenvolvimento de uma aplicação recebem o nome de CASE (Computer-Aided Software Engineering - Engenharia de Software Auxiliada por um Computador).

Os procedimentos são elos de ligação entre as camadas de tecnologia – métodos e ferramentas – de forma a sincronizá-las para obtenção de uma aplicação, software a ser desenvolvido. Esta camada forma a base para o gerenciamento e controle dos projetos de software e estabelece um contexto no qual os métodos serão aplicados, os produtos serão produzidos, a qualidade será mensurada e as mudanças serão gerenciadas (PRESSMAN, 1995, p.23-24).

3.2 DEFINIÇÃO DA ENGENHARIA WEB (WEBE)

Pressman (2002) relata que a engenharia web é um processo para desenvolver uma WebApp de alta qualidade; Ginige (2002) a diagrama como uma arquitetura de conceitos, técnicas de estruturação de dados e gerenciamento de serviços. Baixando um pouco o nível de visão, visualiza-se Rossi e Shwabe (2000) declarando-a como a concepção completa de alguns passos pré-definidos, com escopo determinado, para chegar a um produto de fácil manutenção, agradável e de rápida implementação. Todos com o intuito final de se desenvolver e gerenciar uma WebApp administrável e modularmente flexível.

Para a aplicação de algo tão complexo e detalhado, pode-se levar em consideração a sugestão de Henichker e Koch apud Locatelli (2003), que sugerem dividir esforços para se desenvolver o sistema, porém, fazendo o crescer de forma homogênea. Essa divisão é um desafio, tendo em vista requisitos como imediatismo, segurança e estética. Esses fatores são imprescindíveis quando se trata de uma aplicação que está on-line.

3.3 REQUISITOS PARA A QUALIDADE DE UMA WEBAPP

Seguindo a linha de raciocínio, proposta por Pressman (2002), a mesma qualidade mensurada em sistemas também pode ser mensurada em Webapps. Nesta linha, seis atributos-chave, para a mensuração da qualidade deste tipo de aplicações (on-line), foram destacados pela norma ISO 9126 (1997), com tentativa de identificar os principais atributos de qualidade para software de computador.

Olsina (1999) levanta os seis atributos de qualidade em aplicações web (WebApps), de forma detalhada na seguinte árvore (Figura5):



Fonte: Pressman (2002, p. 756).

Figura 5: Árvore de Olsina para representação de Requisitos de qualidade da WebApp.

Segue abaixo uma breve definição das características de qualidade da aplicação web:

Usabilidade: Facilidade de uso, levando em consideração: a estética, interfaces abstratas, legibilidade de textos e imagens, clareza e concisão das informações expostas, padronização de temas, e atributos visuais que permitem o usuário situar-se em seu ciclo navegacional: títulos, menus e *links*¹⁶ ou âncoras¹⁷ de acesso.

Funcionalidade: Capacidade da aplicação de oferecer serviços de busca e recuperação de informações, navegabilidade, precisão e fácil acesso a informações disponíveis na WebApp.

Confiabilidade: Recuperação e tratamento de erros, suporte a acessos múltiplos sem perda de informações, verificação de erros de links (“links quebrados”, ou seja, com seu destino perdido ou incorreto), validação dos dados de entrada do usuário e dos dados de saída do sistema. Capacidade do servidor manter, no maior tempo possível, on-line os serviços oferecidos aos usuários.

¹⁶ Links são denominações dadas a palavras ou textos que, quando clicados, levam a outras informações, geralmente utilizada para acessar a maiores referências sobre o assunto que foi clicado.

¹⁷ Mesma função que os links, porém feitos sobre imagens.

Eficiência: Desempenho perante a cadênciā do sistema para retornar informações solicitadas pelo usuário, sendo elas: textos, imagens, sons, vídeos, consultas de busca ou acesso a informações em repositórios de dados.

Mantenabilidade: Como WebApps são geralmente orientados a conteúdo, a necessidade de um maior acesso à atualização e manutenção destas informações é atributo primordial para que se mantenha a qualidade desta aplicação.

Portabilidade: Por se tratar de um sistema independente da plataforma utilizada pelo usuário, mas dependente dos navegadores de Internet (Browsers): a portabilidade é um item pouco relevante quanto a mensuração da qualidade de uma WebApp. Esse motivo explica o fato deste atributo não estar presente na árvore de Olsina apresentada anteriormente (OLSINA, 1999); (PRESSMAN, 2002).

Os atributos descritos acima não englobam todas as medidas necessárias para a completa avaliação de qualidade de um software, mas, segundo Pressman (2002, p. 504), eles fornecem uma “base valiosa para medidas indiretas e uma excelente lista de verificação para avaliar a qualidade de um sistema”, ou seja, avaliar o nível de qualidade da WebApp.

3.4 PROCESSO DE DESENVOLVIMENTO

Conforme Pressman (2002) e Molinari (2004), a principal dúvida que inicia o processo de desenvolvimento de uma WebApp é quanto sua importância, e para quem se destina o serviço, idéias que precisam estar claras para que se possa definir itens como:

- tecnologia que melhor se enquadra as necessidades da aplicação;
- níveis de qualidade em relação ao orçamento e prazos disponíveis;
- equipe que irá desenvolver a WebApp;
- e, principalmente, o público alvo desta aplicação.

Segundo uma pesquisa realizada pela revista Epner citada por Ginige (2002), 84% dos sistemas web não atendem as necessidades do mercado; 53% não cobrem suas funcionalidades requeridas; 79% dos atrasos resultam de problemas no projeto; e 63% dos projetos excedem seus custos estimados.

Estes dados estatísticos permitem mensurar a importância do bom esclarecimento quanto as dúvidas iniciais que surgem ao início do desenvolvimento de WebApps, bem como a necessidade de uma sintonia entre a equipe de desenvolvimento, as tecnologias a serem adotadas na aplicação e principalmente ao grande público alvo desta aplicação.

3.5 ETAPAS DO PROCESSO

Para o sucesso de uma WebApp, cada etapa apresentada no modelo exíral (Figura 1) deve ser aplicada e explanada corretamente. Pois entre as atividades que serão adotadas nessas etapas estão a visualização real do problema, a melhor aplicação da tecnologia atual para a resolução desses problemas, e principalmente, o contato inter-pessoal entre os vários papéis existentes na engenharia web para o melhor entendimento e resolução destes problemas. Tomando como exemplo têm-se: os programadores – responsáveis por codificar as soluções em tecnologias eletrônicas; os editores – responsáveis por diagramar as informações contidas nas aplicações; designers – responsáveis pela parte estética e visual das WebApps; os internautas – pessoas que utilizam as aplicações através da Internet; os empresários – responsáveis por bancar os custos existentes no desenvolvimento dessas aplicações e também os principais interessados no retorno destas soluções, entre outros.

3.5.1 Formulação, Planejamento e Análise

O ciclo do modelo incremental (Figura 3) inicia na fase de **formulação**, no qual se define o escopo do primeiro incremento (metas e objetivos da WebApp).

Perguntas fundamentais são levantadas nesta fase:

- Qual o objetivo principal desta aplicação a ser desenvolvida?
- Qual será a utilidade desta aplicação?
- Quais serão os usuários desta aplicação?

As questões acima deverão ser respondidas de forma direta e sucinta. Por meio delas serão identificados as duas categorias de metas:

- **metas de informação:** indicando o objetivo de fornecer a informação específica ao usuário.

- **metas de aplicativo:** indicando quais serão as atividades executadas pelo aplicativo.

Após a identificação das metas descritas acima, será possível identificar o perfil do usuário (background, conhecimento e preferências) para o qual o sistema será desenvolvido (PRESSMAN, 2002; PAIVA, 2003).

Quando todas metas e perfis de usuários estiverem desenvolvidos, o ciclo passa para uma avaliação de riscos e estimativa de custos, o **planejamento**, onde também é realizado o escopo geral do sistema e um cronograma para realização das novas atividades (PRESSMAN, 2002).

O levantamento de requisitos técnicos e de conteúdo, junto com requisitos do projeto gráfico são definidos na **análise**. Nesta fase encontram-se quatro diferentes tipos de análise (PRESSMAN, 2002; PAIVA, 2003):

- **análise de conteúdo:** responsável pela identificação e organização do conteúdo que será disponibilizado na aplicação (textos, imagens, sons e vídeos...);
- **análise de interação:** seqüências de interação entre o usuário e a aplicação descrevendo casos de uso detalhados de alguns componentes;
- **análise funcional:** descrição detalhada de funções e operações; e
- **análise da configuração:** detalhes de ambientes e infra-estrutura na qual a aplicação irá residir.

3.5.2 Projeto

A partir deste momento, as atividades (Figura 3) entram na **engenharia**, onde se dividem em duas linhas de tarefas paralelas:

- Projeto de conteúdo e produção;
- Projeto arquitetural, navegacional e de interface.

Conforme Pressman, apud Breve (2002), o projeto de conteúdo e produção destina-se à elaboração e coleta de informações e artefatos não técnicos. Essas atividades são executadas por equipes, normalmente, pouco vinculadas ao desenvolvimento técnico da

aplicação. Redatores, designers, músicos, fotógrafos, entrevistadores são exemplos de elementos dessa equipe.

Os curtos prazos de desenvolvimento e a rápida e constante evolução do conteúdo destes sistemas exigem dos desenvolvedores a resolução de problemas imediatos, e com os mesmos olhos, devem criar uma arquitetura que suporte uma evolução contínua.

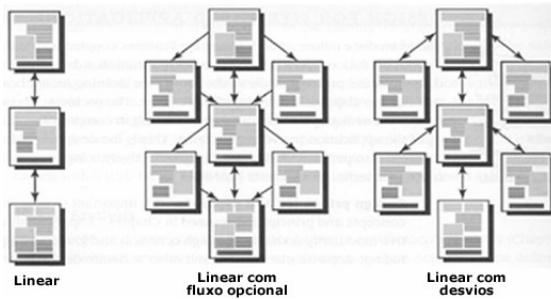
Segundo Pressman (2002) e Paiva (2003), visando um projeto efetivo, faz-se necessário a reutilização de quatro elementos técnicos básicos:

- métodos e Princípios de Projeto: regras de uma modularidade eficiente (alta coesão e baixo acoplamento) são bem vindas no desenvolvimento de uma WebApp, assim como todos os métodos de projetos para sistemas orientados a objetos, permitindo tanto a alta dinâmica e reaproveitamento de objetos do código devido ao baixo acoplamento, bem como, o melhor aproveitamento de atividades já executadas por diversos objetos, permitindo uma alta coesão;
- regras de outro: aplicação de heurísticas no projeto de novas aplicações;
- Design Patterns: padrões genéricos para resolução de problemas comuns que podem adaptar-se a problemas bem mais específicos; e
- modelos (Gabaritos ou Templates): modelos que geram esqueletos para padronização de interfaces de um aplicativo.

Na fase do projeto arquitetural é definida a estrutura que a aplicação irá utilizar, de acordo com as informações adquiridas no Projeto de conteúdo. Esta estrutura irá auxiliar na criação dos modelos (templates) da aplicação, bem como, a forma estrutural em que o conteúdo será apresentado ao usuário (PRESSMAN,2002).

Conforme Dinucci(1998) apud Pressman (2002), há quatro possíveis estruturas:

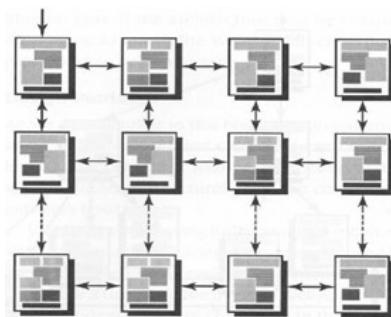
Estrutura Linear: representa uma seqüência de interações previsíveis. Utilizada quando se deseja uma continuidade fixa e direcionada de informações, em que o usuário visualiza opções diretas e inflexíveis (Avançar, voltar ou “escolha uma opção”), como por exemplo tutorias ou preenchimento de pedido de produtos.



Fonte: Pressman (1995, p. 762).

Figura 6: Estruturas Lineares.

Estrutura de Malha: representa uma estrutura de duas ou mais camadas. Utilizada normalmente para o cruzamento de informações destinadas a um mesmo fim, por exemplo, uma loja de discos, onde o usuário pode procurar o produto por relação de compositores, ou por gêneros musicais, ou ainda por promoções do dia. Todos cruzamentos irão chegar ao mesmo destino: a descrição de um disco em específico.

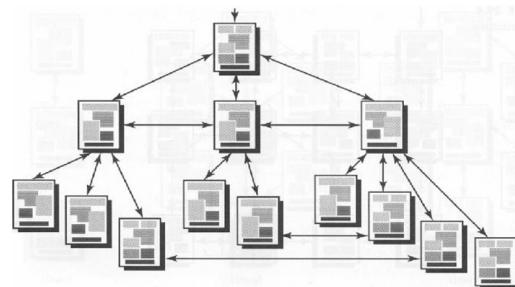


Fonte: Pressman (1995, p. 763).

Figura 7: Estrutura em Malha.

Estrutura Hierárquica: sendo a mais comum das arquiteturas, esta estrutura se assemelha muito com um grafo de árvore, porém permite que não somente as camadas pais e filhos se comuniquem, mas também que haja comunicação entre as camadas irmãs. Utilizada na maioria dos sites, segue o esquema tradicional, por exemplo, possui a *home page* (página inicial), na qual se encontram os links com as páginas de informações sobre a empresa, de produtos e de contato. Dentro de cada dessas sub-páginas existe um link para suas páginas irmãs e também para suas páginas filhas, como, por exemplo, dentro da página de empresa terá um link para fotos da empresa, institucional, equipe, atuação, mas também para produtos e

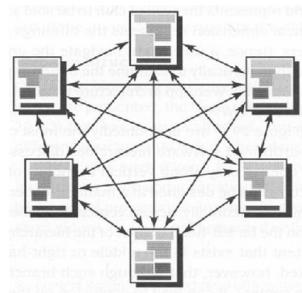
contatos, levando em consideração que os links pertinentes a suas páginas irmãs só sejam vistos no momento que essas forem acessadas.



Fonte: Pressman (1995, p. 764).

Figura 8: Estrutura Hierárquica.

Estrutura Interligada: representa uma estrutura altamente interligada, muito similar à maneira de representação de uma arquitetura de sistemas orientados a objeto. Essa arquitetura permite que cada página tenha capacidade de se comunicar, dinamicamente com qualquer outra página pertinente ao sistema, proporcionando uma alta flexibilidade da aplicação, porém com alta probabilidade de confundir o usuário.



Fonte: Pressman (1995, p. 764).

Figura 9: Estrutura Interligada ou Pura teia.

Esses modelos podem ser trabalhados separadamente ou em conjunto, de acordo com as necessidades da aplicação.

No projeto navegacional identifica-se os caminhos que permitirão aos usuários o acesso a conteúdos e serviços pré-definidos para suas permissões.

Pressman (2002), citado por Breve (2002), estabelece uma ou mais unidades semânticas de navegação (*semantic navigation unit* – SNU) para cada tarefa permitida a cada tipo de usuário existente. Esta estrutura SNU é formada de várias sub-estruturas navegáveis chamadas de caminhos (*ways of navigating* – WoN). Cada WoN será responsável por guiar o usuário até seu objetivo ou grupo de tarefas. A WoN também possui uma composição de nós relevantes (*navigational nodes* – NN) ligados por *links* (*navigational links*), que, por sua vez, podem possuir outras SNUs.

O projeto de interface utilizada para aplicações web utiliza os mesmos conceitos da engenharia de software tradicional, porém com alguns detalhes especiais. (PRESSMAN, 2002)

A interface de aplicações web também é a primeira impressão de um sistema ou organização disposta na Internet. Seu grau de responsabilidade por manter as aparências é bastante relevante. Conforme Nielsen (2000), pode-se seguir algumas recomendações simples, que garantem uma boa interface:

- reduzir ao máximo os erros no servidor, mesmo os mínimos. Podem gerar desconforto ao usuário, fazendo-o buscar informações em outro lugar;
- objetividade nos textos, evitando forçar com que o usuário leia muita quantidade de texto, principalmente explicações sobre naveabilidade e funções do sistema;
- evitar a utilização de avisos “Em desenvolvimento!”. Eles provocam expectativa e ansiedade no usuário, podendo decepcioná-lo;
- exibir as informações principais no topo das páginas, evitando que os usuário usem a barra de rolagem, comumente indesejáveis;
- nunca contar somente com as funcionalidades do navegador. Projetar menus de navegação consistentes, que permitam a completa navegação do usuário; e
- manter bem expressiva as opções de navegação, não permitindo a busca inesperada de *links* dentro de uma página.

Uma interface, para permitir uma melhor aceitação do usuário, não precisa necessariamente ser deslumbrante, mas deve possuir uma boa estrutura e ser

ergonomicamente sólida, permitindo ao usuário identificar sua localização atual no sistema, o que ele pode fazer e como fazer o que ele quer e pode fazer (PRESSMAN, 2002),.

3.5.3 Implementação e Testes

Após a fase da engenharia, efetua-se a construção (implementação) dos itens levantados e planejados. Nesta fase, as páginas são criadas e seus conteúdos são revisados e testados.

As atividades de testes em aplicações web seguem o mesmo objetivo que a engenharia convencional: encontrar erros. Pressman (2002) e Paiva (2003) destacam algumas atividades, recomendadas para sistemas orientados a objetos, que podem auxiliar na detecção de erros:

- **revisão do modelo de conteúdo:** objetiva encontrar erros de tipografia e gramática, consistência do conteúdo, representações gráficas, entre outros, relacionados ao valor teórico das informações postadas na WebApp.
- **revisão de links e navegação:** o sistema é exaustivamente navegado, com vários usuários, com diferentes permissões, a fim de detectar erros de links ou visualização indevida de informações.
- **processo de teste de unidades:** cada página é testada individualmente, procurando por erros de scripts, carregamento de imagens ou outras mídias.
- **testes de integração:** procura pela conectividade e comunicação entre as páginas da aplicação. Gerando seqüências de SNUs pré-definidas, verifica se a colaboração de funções auxiliares estão realmente sendo solicitadas e, se seus resultados permitem a real execução das tarefas desejadas.
- **teste de funcionalidade geral e conteúdo fornecido:** averigua a consistência dos dados gerados pelo sistema através de modelos reais, corrigindo erros de integridade das informações.
- **teste de ambientes e configurações:** a WebApp é visualizada e manipulada em vários tipos de navegadores, plataformas, hardwares e conexões com a Internet procurando encontrar erros associados a cada uma das possíveis configurações.

- **teste por população controlada:** são simuladas interações de cada tipo possível de usuários no sistema, a fim de encontrar erros de conteúdo e navegação, desempenho e confiabilidade da WebApp, bem como questões de usabilidade e compatibilidade.

3.5.4 Avaliação do Cliente

Concluído a fase de implementação, o contato do cliente é requisitado, para que seja possível a **avaliação das atividades** já executadas.

Nesta fase do processo pode haver as seguintes atividades (PRESSMAN,2002):

- a ocorrência de alterações no escopo;
- a confirmação ou alteração de usuários e suas permissões;
- a reedição da interface, bem como, do projeto navegacional.

Esses levantamentos deverão ser repassadas novamente para faze de formulação, porém essa ação dependerá da disponibilidade tanto financeira como de prazos disponibilizados a esta aplicação.

4 METODOLOGIA OOHDM

Muitas são as opções para escolha de uma metodologia de desenvolvimento de WebApps, como visto no capítulo 2.2 (Modelos e Metodologias de Desenvolvimento Multimídia), porém há uma que se encaixa perfeitamente nos requisitos de processo propostos por Pressman (2002, p.756), onde ele cita que “o imediatismo e a evolução continuada”, necessários para o desenvolvimento de WebApps, “ditam um modelo de processo interativo e incremental”.

Para justificar a escolha da metodologia OOHDM, utiliza-se uma das conclusões da pesquisa do professor Costagliola (2002), onde ele cita que não existe uma *the best*¹⁸ metodologia, mas sim, que existem metodologias específicas para diferentes focos que se deseja aplicar sobre o desenvolvimento de uma aplicação hipermídia. E dentre as metodologias apresentadas no estudo do professor Costagliola (2002) o OOHDM foi considerado uma metodologia bem aplicada ao desenvolvimento de grandes aplicações hipermídia, através de um processo de desenvolvimento incremental baseado em prototipação, essa prototipação para Pressman (2002) significa uma maior interatividade entre o desenvolvedor e o cliente. O que permite concluir que o OOHDM é a metodologia ideal para o desenvolvimento de WebApps através da engenharia proposta por Pressman (2002).

4.1 O OOHDM

Pode-se visualizar aplicações hipermídia como sistemas projetados para funcionar como parte de uma equipe homem-máquina. A parte do problema a ser resolvido pela máquina usa técnicas apropriadas: bases de dados, sistemas baseados em conhecimento, hipermídia, sistemas de recuperação de informações, entre outros. A parte do problema a ser resolvido pelo homem usa uma estrutura de hipermídia que auxilia na integração transparente destas representações do conhecimento.

Um problema significativo é desenvolver aplicações de hipermídia e fazer com que se integre na estrutura descrita acima.

As aplicações de hipermídia incluem normalmente informações complexas e podem permitir comportamentos sofisticados de navegação. O modelo OOHDM permite uma

¹⁸ A melhor.

descrição concisa de artefatos (complexos de informação) em um roteiro padronizado de navegação e de transformações complexas de interface (SCHWABE, 2004).

A modelagem OOHDMD (*Object Oriented Hypermedia Design Method* – Modelo de Projeto Hipermídia Orientado a Objetos) é um método que auxilia no desenvolvimento de aplicações hipermídia em larga escala, sendo elas CD-ROM, quiosques, WebApps entre outras (LOCATELLI, 2003).

Segundo Schwabe e Vilain (2002, p. 3) muitas das atividades modeladas no OOHDMD se baseiam na notação proposta pela UML (*Unified Modeling Language*), notações estas encontradas com mais detalhes no site da Rational UML¹⁹ (2004).

A orientação a objetos (OO) utilizada no modelo OOHDMD surgiu na década de 70 e teve como primeira linguagem de programação o SMALLTALK. Essa forma de modelagem OO possui como objetivo, conceituar todo escopo do sistema a ser desenvolvido como um conjunto de objetos que possuam suas próprias características (atributos) e comportamentos (operações ou métodos), conforme comentado por Furlan (1988) apud Hennrichs (2000, p.21).

O HDM, primeiro modelo amplamente conhecido para aplicativos hipermídia baseava-se no modelo entidade-relacionamento. Esse modelo era simples e não possuía muitas primitivas de modelagem, fazendo com que muitas informações referentes ao projeto não fossem documentadas. O que possibilitava somente o desenvolvimento de uma aplicação conceitual ou uma implementação do modelo criado (ROSSI, 1997 apud HENNICHRS, 2000, p. 28).

¹⁹ Página pertencente a IBM, com acesso em: <<http://www.rational.com/uml>>

Sucessor do modelo HDM, o OOHDM manteve algumas idéias de seu precursor e alterou outras, como na tabela abaixo:

Tabela 2: Tabela de Características herdadas e criadas no OOHDM

Características Herdadas do HDM	Características Criadas ou Enriquecidas
Reconhecimento de que a modelagem é independente do ambiente e do modelo de referência.	Não é apenas uma abordagem de modelagem, mas sim uma metodologia, pois aborda muitas atividades.
Reforça as estruturas hierárquicas, oferecendo a possibilidade de construção de agregados.	Na fase de modelagem conceitual as primitivas são mais ricas. Como são orientada a objetos suportam: agregação, generalização e herança.
Inclui o conceito de perspectiva de atributo, onde cada classe folha implementa uma visão possível do atributo.	É possível fazer uso do esquema conceitual para definir perfis de usuários diferentes.

Fonte: Schwabe (apud LOCATELLI, 2003, p. 21).

Na modelagem OOHDM, uma aplicação hipermídia é desenvolvida em cinco etapas: levantamento de requisitos, modelagem conceitual, projeto navegacional, projeto de interface abstrata e implementação, permitindo uma mistura de modelo de processos incremental e de prototipação.

As quatro primeiras etapas focam um contexto de projeto e geram um modelo de desenvolvimento orientado a objetos em particular. Composição, agregação e generalização/especialização são usadas para conseguir um maior poder de abstração e reuso dos componentes.

A quinta etapa, a atividade de implementação, somente será desenvolvida após o término das quatro primeiras etapas (SCHWABE, 2002).

O quadro a seguir resume cada etapa, com seus devidos produtos, mecanismos e objetivos no OOHDM:

Quadro 1: Esboço da metodologia OOHDM

Etapas	Produtos	Mecanismos	Objetivos
Levantamento de Requisitos	<ul style="list-style-type: none"> • Atores e Tarefas • Use Cases • UID's 	<ul style="list-style-type: none"> • Entrevistas e análise do negócio • Cenários e análise dos Use Cases 	Capturar as exigências e requisitos para a aplicação
Modelagem Conceitual	<ul style="list-style-type: none"> • Classes, com seus devidos atributos e operações. • Subsistemas • Relacionamentos • Perspectivas de atributos 	<ul style="list-style-type: none"> • Classificação • Agregação • Generalização • Especialização 	Modelar a semântica do domínio da aplicação
Projeto Navegacional	<ul style="list-style-type: none"> • Nós e Elos • Estruturas de acesso • Contextos de navegação • Transformações navegacionais 	<ul style="list-style-type: none"> • Mapeamento entre objetos conceituais e de navegação • Padrões de navegação para a descrição da estrutura geral da aplicação. 	Leva em conta o perfil do usuário e a tarefa, com ênfase em aspectos cognitivos e arquiteturais.
Projeto de Interface Abstrata	<ul style="list-style-type: none"> • Objetos de interface abstrata • Comportamentos a eventos externos • Transformações de interface. 	<ul style="list-style-type: none"> • Mapeamento entre objetos de navegação e objetos de interface. 	Modelagem de objetos perceptíveis, implementação de metáforas escolhidas, descrição da interface para os objetos navegacionais.
Implementação	<ul style="list-style-type: none"> • Aplicativo em execução 	<ul style="list-style-type: none"> • Aqueles que o ambiente alvo fornecer. 	Desempenho, completitude.

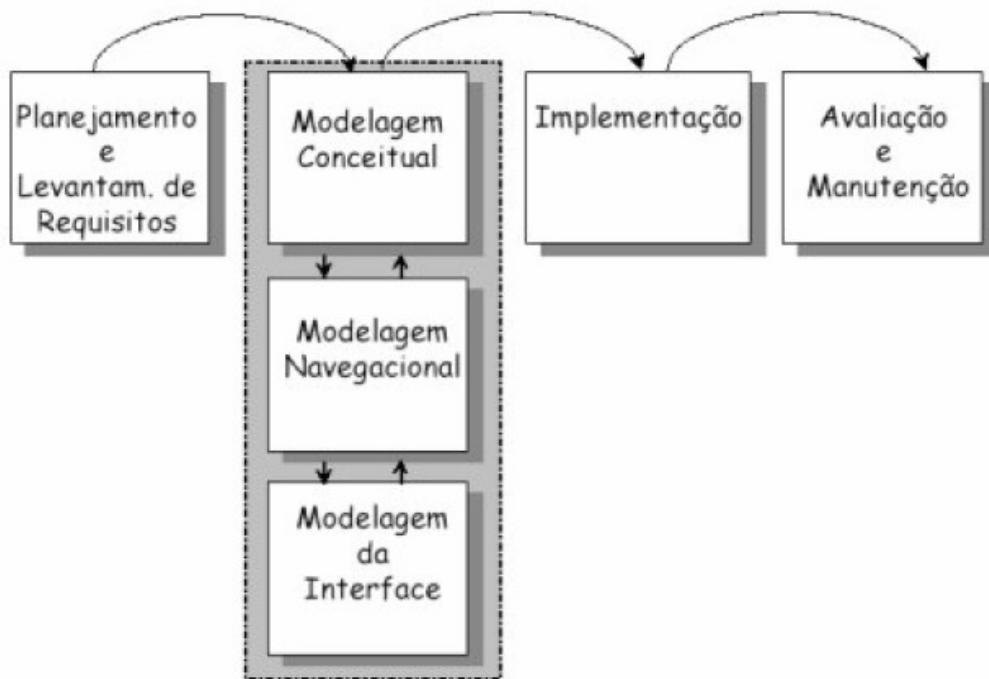
Fonte: Schwabe (2004).

As atividades executadas no OOHDM são iterativas, o que nos permite, segundo Locatelli “realizar uma atividade em paralelo com outra e retornarmos para qualquer atividade caso for necessário” (2003, p. 23).

Locatelli ainda explica que

Utilizando-se a metodologia OOHDM é possível desenvolver projetos modulares e de fácil manutenção, pois a modelagem conceitual, navegacional e o projeto de interface são tratados como atividades separadas o que nos permite concentrar-se em diferentes interesses, resolvendo um a um e retornando para cada etapa construída quando for necessário, com isso obtemos também uma estrutura para o raciocínio sobre o processo de todo o projeto, obtendo experiência específica para cada atividade. OOHDM também oferece independência na escolha de linguagens e ambientes de programação fazendo com que a escolha do projeto seja mais ampla pois pode-se tanto trabalhar com o paradigma de orientação a objetos como programação estruturada, ou as duas ao mesmo tempo (2003, p. 23).

A modularização dos projetos citada por Locatelli se torna visível na figura 5 apresentada por Hennrichs (2005) para representação do ciclo de desenvolvimento utilizado pelo OOHDM.



Fonte: Hennrichs (2005).

Figura 10: Ciclo de desenvolvimento usando OOHDM.

A seqüência e interação de atividades propostas pelo OOHDM ficam visíveis na figura abaixo, em uma adaptação a imagem original de Pressman e Rossi (Figura 34) presente no Capítulo 6.

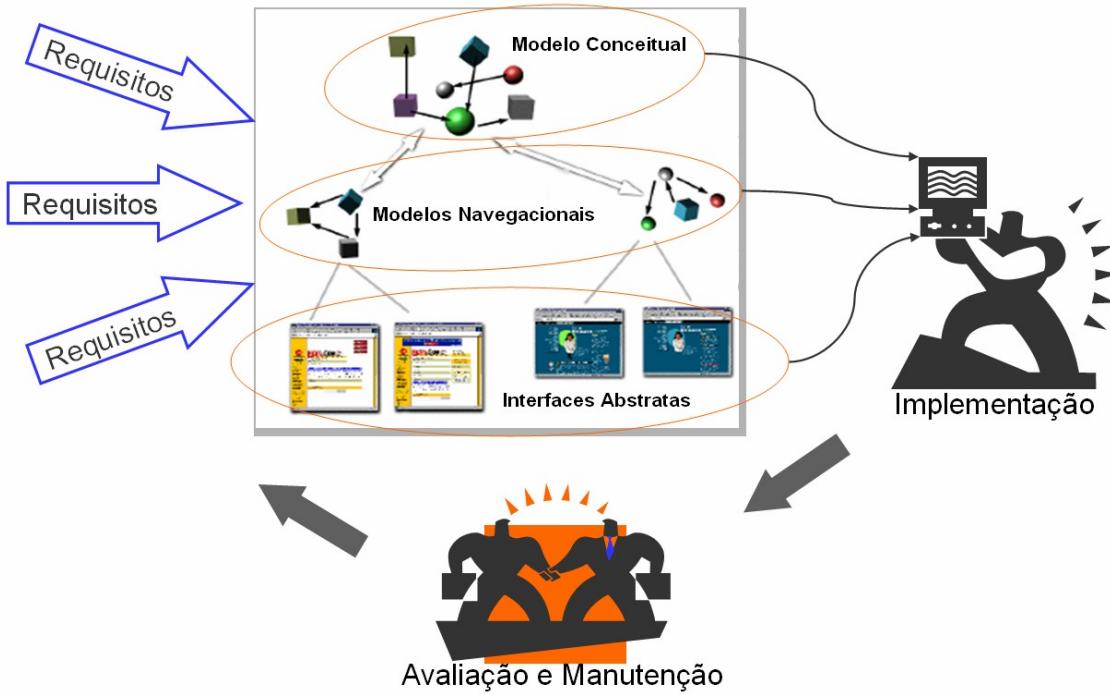


Figura 11: Ciclo de desenvolvimento usando OOHDM.

Na figura acima, uma gama de requisitos, tanto funcionais como não funcionais, são coletados através de estudos sobre o assunto a ser focado, como em entrevistas e necessidades coletadas pelos futuros usuários do sistema. Esses requisitos são documentados e aprovados por esses usuários, gerando a partir desse momento um conjunto de três níveis de modelagens: conceitual, navegacional e de interface abstrata. Concluídas estas modelagens, as mesmas são utilizadas para a geração do software propriamente dito. A medida que este software vai sendo desenvolvido, novas entrevistas e acordos vão sendo definidos entre a equipe de desenvolvimento e os usuários do sistema, nestes acordos são levantados novos requisitos que voltam a seguir todo ciclo novamente, até chegar ao objetivo final esperado pela aplicação.

4.2 LEVANTAMENTO DE REQUISITOS

A fase de levantamento de requisitos tem por objetivo definir quais serão os usuários da aplicação e quais tarefas deverão ser apoiadas, bem como todos os requisitos necessários para estabelecer o domínio da aplicação do problema (Hennrichs, 2005).

Para Hennrichs (2005) a fase de levantamento de requisitos está dividida em cinco etapas, explicadas por Schwabe e Vilain (2002, p.04) como:

(1^a) Identificação de Atores e Tarefas: os atores são todos os elementos que irão acessar a aplicação através do papel de um usuário do sistema. Sua identificação é importante, pois nem sempre os atores participarão do mesmo cenário e, sendo assim, são interagidos somente os cenários referentes a classe de atores a ele pertinentes. As tarefas são os objetivos que os atores pretendem alcançar ao interagir com a aplicação.

(2^a) Especificação dos Cenários: é uma descrição detalhada das tarefas que o ator deseja executar dentro do domínio da questão. Para isso, deve-se inicialmente identificar os atores que executarão a tarefa e a quais classes eles pertencem. Para cada classe o ator deverá possuir uma identificação exclusiva. Esta descrição é feita de forma textual pelo projetista ou pelo usuário.

(3^a) Especificação dos Use Case (Casos de Uso): Segundo Rumbaugh (1999 apud SCHWABE & VILAIN, 2002, p. 05-06), um Caso de Uso é apenas uma maneira de usar o sistema, abordando somente informações visíveis para o usuário, sem descrições do funcionamento interno da aplicação. Para Rumbaugh (1994) e Jacobson (1995), citados por Schwabe e Vilain (2002, p.05-06), pode-se dizer que um cenário nada mais é que uma instância de um Caso de Uso. Desta forma, é necessário primeiramente reunir todos cenários que diagramam a mesma tarefa para que se possa generalizar um Caso de Uso para aquela tarefa.

(4^o) Especificação do UIDs (*User Interaction Diagram*²⁰): visa uma representação gráfica dos Use Cases com o objetivo de apresentar as interações entre o usuário e o sistema sem detalhes referentes a interface com o usuário.

(5^a) Validação dos Use Cases e UIDs: apresenta-se todos os Use Cases e UIDs desenvolvidos aos usuários do sistema, a fim de estabelecer de forma clara, a interação destes, com o sistema, buscando um consenso comum entre as reais necessidades do usuário e a forma de solucioná-la.

²⁰ Diagrama de Interação do Usuário.

Após a conclusão destas cinco etapas, tem-se em mãos o material necessário para proceder às próximas fases do OOHDMD.

4.3 MODELAGEM CONCEITUAL

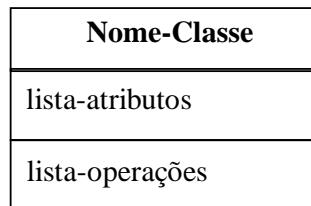
O modelo Conceitual consiste em “analisar o domínio da aplicação, obtendo todas as informações importantes para o desenvolvimento do projeto” (LOCATELLI, 2003, p. 23), desenvolvendo um esquema conceitual que consistem em “representar objetos e relacionamentos existentes no domínio da aplicação” (HENNRICHS, 2005).

4.3.1 Componentes do Esquema Conceitual

Conforme Shwabe e Vilain (2002) a modelagem conceitual gerará um esquema conceitual composto por classes, relacionamentos e subsistemas, ou seja, o domínio da aplicação.

As classes de um esquema conceitual objetivam representar um conjunto de entidades que possuam as mesmas características (atributos, relacionamentos e métodos).

Seu corpo é dividido em três partes de forma horizontal. Na primeira parte, insere-se o nome da classe junto com suas propriedades gerais, quando existirem. Na segunda parte, são inseridos os atributos pertinentes a esta classe. E no ultimo campo, adiciona-se a relação de operações possíveis de execução pelas instâncias desta classe.



Fonte: Shwabe e Vilain (2002, p. 23).

Figura 12: Representação de uma classe.

Segundo Shwabe e Vilain (2002) as partes da classe podem ser descritas da seguinte forma:

- **Nome da Classe:** deve ser escrito em negrito e centralizado, com a primeira letra em maiúsculo. As propriedades são responsáveis por informar dados

adicionais à classe, como autor ou data da ultima atualização, e devem estar sempre entre “{” e “}”. Caso a classe seja abstrata, o nome deve apresentar-se também em itálico.

- **Atributos da classe:** representam as propriedades dos objetos, sendo que cada atributo deverá possuir um tipo predefinido simples, complexo ou um tipo definido pela aplicação, este tipo do atributo é considerado como perspectiva do atributo. O atributo pode ser apresentado com a seguinte definição:

visibilidade nome : perspectiva = valor-default²¹ {propriedade}

- **Métodos ou operações da classe:** são operações e manipulações sobre as informações da aplicação. Possuem seus nomes iniciados em minúsculo. A operação pode ser apresentada com a seguinte definição:

Visibilidade nome (lista-parâmetros): expressão-resultado {propriedade}

A figura seguinte representa uma classe de um esquema conceitual:

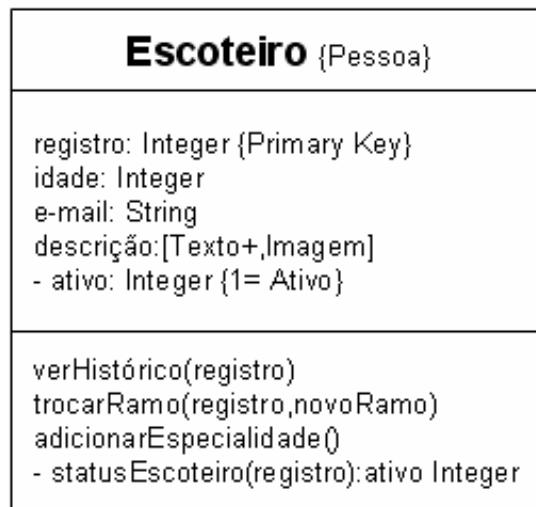


Figura 13: Exemplo de uma classe do esquema conceitual.

²¹ Default é uma representação em inglês do termo defeito, mas, para este caso, significa valor padrão, na idéia de que não haja um outro valor válido para aquela informação.

Na figura 13 é possível identificar todos os componentes de uma classe do esquema conceitual, como:

- Nome da Classe: Escoteiros, tendo um parâmetro de observação {Pessoa} que indica a classe de origem;
- A relação de atributos possui os seguintes itens: registro com perspectiva Integer, tendo como observação que é a chave primaria de identificação dos elementos classe, idade com perspectiva Integer, e-mail com perspectiva String, descrição podendo conter como perspectiva Texto como obrigatório, devido ao sinal “+”, e uma Imagem como opcional e um atributo oculto, devido ao sinal “-“ em sua frente, chamado ativo com perspectiva Integer e uma observação de quanto contiver o valor 1 significará Ativo;
- A lista de operações com os itens: verHistórico, que necessita do parâmetro registro, trocarRamo, que solicita os parâmetros registro e novoRamo, adicionarEspecialidade e um método oculto chamado statusEscoteiro, que utiliza o parâmetro registro para retornar o atributo ativo do tipo Integer.

“A generalização/especialização é utilizada para denotar que classes especializadas herdam todas as características (atributos, métodos e relacionamentos) de classes generalizadas, podendo acrescentar novas características” (SCHWABE & VILAIN, 2002, p. 29).

A representação da generalização/especialização é demonstrada na ligação, através de uma linha, da classe pai (classe generalizada) à classe filha (classe especializada), em que a classe pai é identificada por possuir, na sua ligação com a linha, uma seta vazada, conforme a figura abaixo (SCHWABE & VILAIN, 2002):

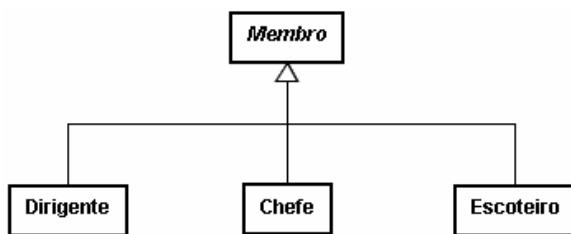


Figura 14: Exemplo de generalização/especialização.

De acordo com a figura 14, o Membro representa a classe generalizada e o Dirigente, Escoteiro e o Chefe as classes especializadas.

Na modelagem de um sistema, determinados objetos estarão relacionados e estes relacionamentos necessitam de uma modelagem para maior clareza. Na UML há cinco tipos de relacionamento. Como o OOHDML utiliza-se da notação UML, ele possui três destes cinco tipos, sendo eles o relacionamento unário, binário e o ternário, como sugere o exemplo seguinte (UML, 2003; SCHWABE & VILAIN, 2002):

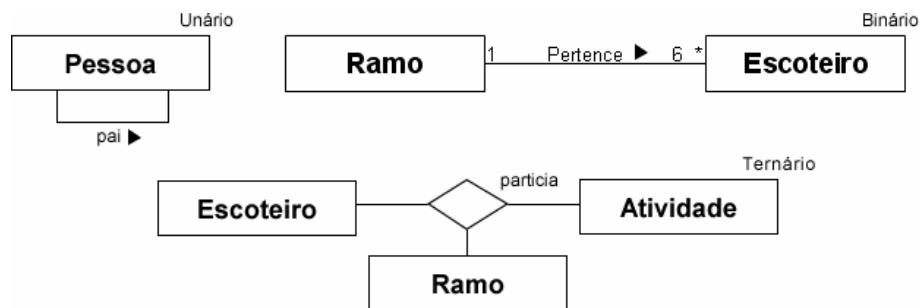


Figura 15: Exemplo de Relacionamentos.

Observa-se também, na Figura 15 apresentada anteriormente, uma descrição do relacionamento junto à linha e, em dois dos modelos, há ainda um triângulo preto preenchido, o qual sugere o fluxo do diagrama.

Além da descrição do relacionamento, outro item fundamental é a cardinalidade do relacionamento. A cardinalidade é responsável por definir o grau de relação entre as classes de um relacionamento, conforme a tabela seguinte:

Quadro 2 : Graus de cardinalidade em um relacionamento

0..1	- a cardinalidade varia entre 0 ou 1
1	- a cardinalidade é somente 1
0..*	- a cardinalidade varia de 0 até infinito
*	- a cardinalidade varia de 0 até infinito
1..*	- a cardinalidade varia de 1 até infinito

1..6	- a cardinalidade varia de 1 até 6
1..5,10..15,20..*	- a cardinalidade varia de 1 até 5 e 10 até 15 e 20 até infinito

A cardinalidade é inserida nos extremos da linha que está fazendo o relacionamento das classes. A Figura 16 mostra um exemplo entre a classe Ramo e Escoteiro, na qual o relacionamento “pertence” possui as cardinalidades 1 para 6 ou muitos, de que um ramo deve possuir seis ou mais escoteiros.

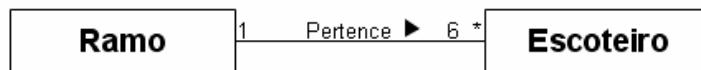
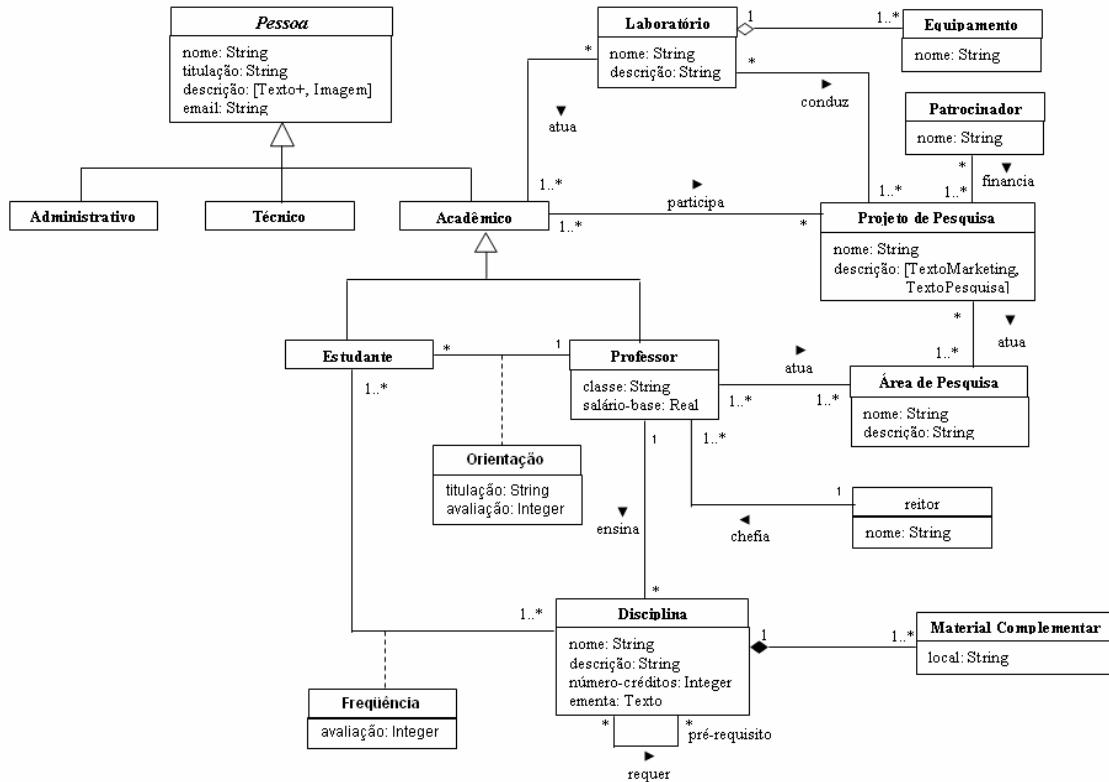


Figura 16: Relacionamento 1-N.

4.3.2 Esquema Conceitual



Fonte: Schwabe & Vilain (2002).

Figura 17: Esquema Conceitual de um Domínio Acadêmico

O esquema conceitual apresentado acima reproduz todas as classes e relacionamentos existentes no domínio acadêmico.

Percebe-se que são definidas as agregações, bem como as composições. A generalização/especialização fica explícita no relacionamento da classe **Pessoa** às classes: **Administrador**, **Técnico** e **Acadêmico**.

Os relacionamentos possuem cardinalidade bem definida e, em alguns casos, é apresentados seus nomes e a seta indicando a direção que este nome deve ser lido.

4.4 MODELAGEM NAVEGACIONAL

De acordo com Schwabe e Vilain (2003, p. 42), Locatelli (2003, p. 27) e Rossi (1997) citado por Hennrichs (2000), a modelagem navegacional irá definir as informações a serem apresentadas aos usuários, bem como, os relacionamentos entre estas relações (navegação). Para cada modelo conceitual pode-se ter um ou mais modelos navegacionais. Caso haja muitos modelos, estes serão específicos para cada tipo de usuários. Essa mudança de modelagem conceitual para modelagem navegacional pode ser identificada, criando-se os **nós** a partir das classes e os **elos** a partir dos relacionamentos.

Hennrichs (2005) apresenta uma subdivisão destas etapas:

- Esquema de Classes Navegacionais;
- Esquema de Contextos Navegacionais;
- Esquema de classes em contexto e
- Cartões de identificação dos objetos criados.

4.4.1 Esquema de Classes Navegacionais

É a definição do conjunto de **nós** e **elos**, presentes em uma visão navegacional de uma aplicação. De acordo com a quantidade de visões existentes na aplicação, pode-se extrair um ou mais esquemas navegacionais (SCHWABE & VILAIN, 2002, p. 42).

Vilain e Schwabe descrevem a representação do esquema conceitual da seguinte forma:

O esquema navegacional é um diagrama onde os nós são representados por retângulos e os elos por linhas, assim como no esquema conceitual. Entretanto, as linhas que representam os elos são direcionadas representando a possível navegação entre as informações disponíveis (2002, p. 42).

Componentes de um esquema navegacional, conforme Schwabe e Vilain (2002):

Nós: seguindo o mesmo conceito de representação das classes do modelo conceitual, os nós diferenciam-se apenas por dois itens: 1) sua função: representação de uma ou mais instâncias de classes do esquema conceitual; e 2) uma característica marcante:

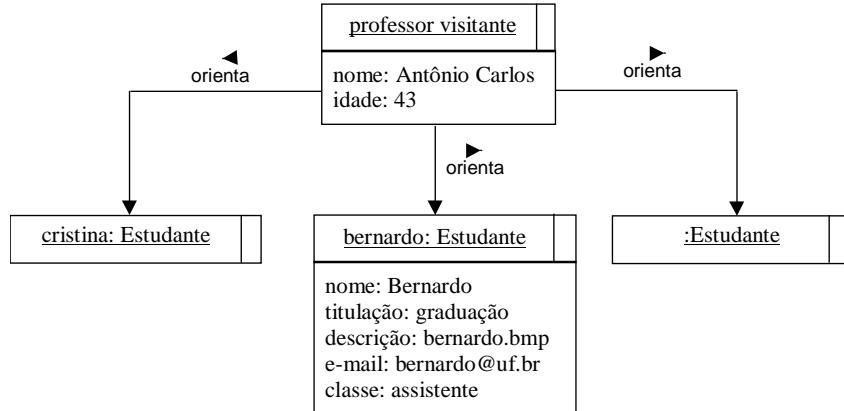
uma linha no lado direito, dividindo verticalmente o primeiro campo de sua representação, onde se encontra o nome.

Elos: São relacionamentos que fazem a conexão entre os objetos navegacionais (nós). Seguem praticamente as mesmas regras de representação existentes nos relacionamentos conceituais, exceto por:

- Possuem somente conexões binárias, relacionando dois nós distintos, e representações unárias, relacionando um nó a ele mesmo.
- Uma seta ao final de sua linha, identificando a navegação permitida a partir do elo. Navegações unidirecionais, com a seta somente em um extremo, indicam a navegação somente no fluxo indicado, e as navegações bidirecionais, com as setas nos dois extremos da linha, indicam a navegação recíproca.
- Os atributos com várias perspectivas, presentes no modelo conceitual, devem ser separados no modelo navegacional, a fim de permitir a visualização do atributo como obrigatório (atributos sem sinais indicam um mapeamento *default* de obrigatoriedade) ou opcional (atributo acompanhado de um asterisco “**”)
- A adição de âncoras e índices junto aos atributos. Elementos responsáveis pela referênciação de contextos e outras informações.

Âncoras e Índices: são responsáveis por “permitir o acesso a um índice ou a um elemento dentro de um contexto” (SCHWABE & VILAIN, 2002, p. 45).

Com a mesma função de um diagrama de objeto na modelagem conceitual, o diagrama de objetos da modelagem navegacional apresenta as instâncias relacionadas de acordo com o diagrama de classes navegacionais definido. Também representa situações, nas quais podem existir objetos com padrões controversos aos das classes definidas no modelo conceitual. Neste caso, o objeto apresenta somente a relação genérica com os outros objetos, sendo representado diretamente no diagrama de classes navegacionais.



Fonte: Vilain (2002, p. 34).

Figura 18: Diagrama de objetos.

Rossi (1997), citado por Hennrichs (2005) explica que o esquema de classes navegacionais apresenta como produto final o diagrama do esquema de classes navegacionais, baseado no esquema conceitual, a figura 19 apresenta um exemplo de diagrama do esquema de classes navegacionais.

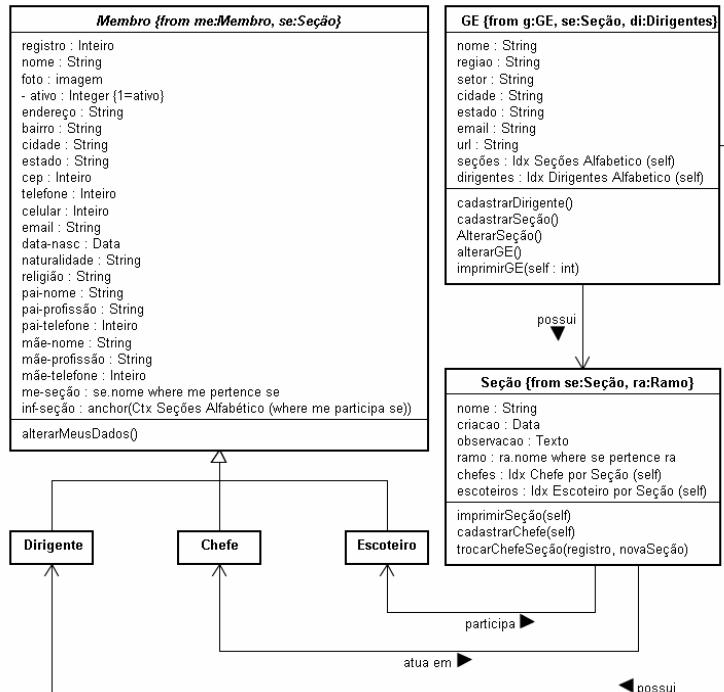


Figura 19: Exemplo de um Diagrama Navegacional.

4.4.2 Esquema de Contexto de Navegação

Tem como objetivo definir em quais contextos a navegação entre as informações será permitida e quais informações serão apresentadas (SCHWABE & VILAIN, 2002, p. 55).

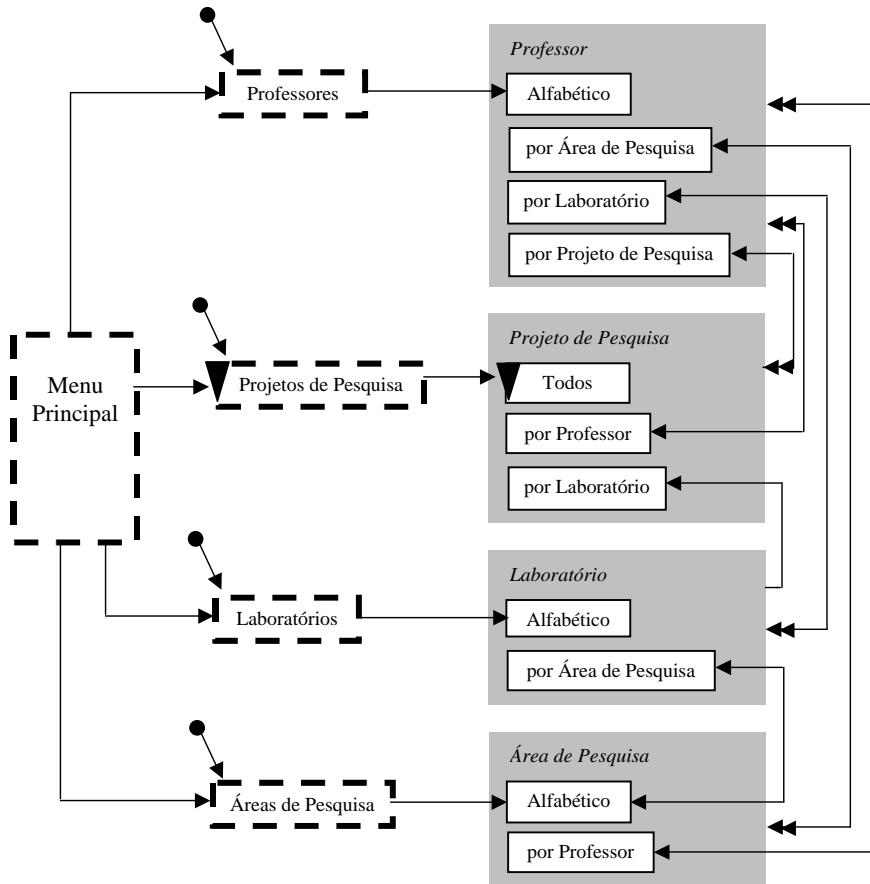
Segundo Hennrichs,

Um Contexto Navegacional é definido como um conjunto de nós, elos e outros contextos navegacionais que auxiliam na organização dos objetos navegacionais. Os nós contêm o domínio da informação (modelos conceitual) e os elos representam os relacionamentos. A construção de contextos navegacionais fornece espaços de navegação consistente, deixando o leitor à vontade em sua navegação pela aplicação e diminuindo as chances dele “perder-se no hiperespaço” (2000, p.44).

O contexto navegacional parte das classes navegacionais, definindo as classes navegacionais com o objetivo de exibir o relacionamento entre os objetos e o esquema navegacional com exibindo assim a navegação em geral (HENNRICHS, 2000, p. 44).

O contexto navegacional é, segundo Shwabe e Vilain (2002), composto por um conjunto de objetos que se relacionam a partir de alguns aspectos em comum. Este contexto pode conter objetos de uma ou mais classes navegacionais e estes objetos podem participar de um ou mais contextos.

A Figura 20 vislumbra um exemplo de um diagrama do esquema de contexto navegacional de uma página acadêmica.



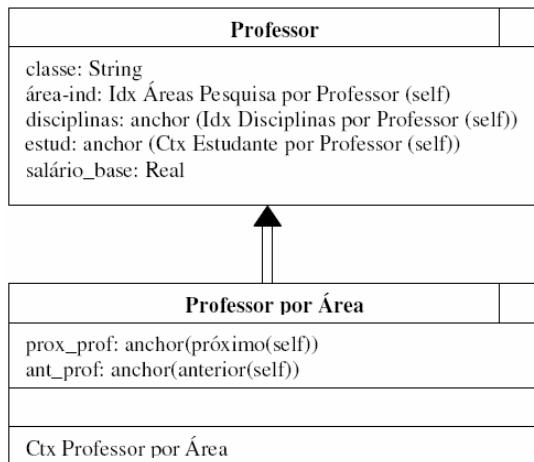
Fonte: Shwabe e Vilain (2002, p. 62).

Figura 20: Esquema de contexto de navegação.

Na figura 20 estão sendo apresentados quatro classes navegacionais, sendo elas: **Professor**, **Projeto de Pesquisa**, **Laboratórios** e **Área de Pesquisa**. O quadro contendo o **Menu Principal**, bem como, todos os quadros delimitados por uma borda grossa e tracejada (**Professores**, **Projetos de Pesquisa**, **Laboratórios** e **Áreas de Pesquisa**) são identificados como estruturas de acesso, sendo estes os responsáveis pelo acesso aos contextos. Os elementos do contexto são identificados como caixas de texto brancas presentes em cada classe navegacional (caixas em cinza) e são responsáveis pelas formas de exibição dessas classes navegacionais. A seta com um círculo preenchido em uma das extremidades e que aponta para uma estrutura de acesso indica que esta estrutura pode ser acessada de qualquer ponto da aplicação e as setas que ligam vários contextos a outros contextos ou classes navegacionais indicam uma navegação direta entre eles e, caso haja setas bidirecionais, é permitida também o retorno ao objeto de origem (SHWABE & VILAIN, 2002).

4.4.3 Esquema de Classes em Contexto e Cartões de Identificação

Segundo Schwabe (2003), citado por Hennrichs (2005), os esquemas de classes em contexto se fazem necessários quando algum respectivo nó apresenta uma anomalia no padrão de apresentação, fazendo-se necessária a utilização de uma classe específica para este contexto. A representação deste tipo de atividade se dá na apresentação de uma classe subsequente a classe normal do contexto tratado. Esta classe é dividida em quatro partes, sendo igual a classe padrão, porém mantendo a terceira parte para a especificação das novas operações desta classe. Como mostra a figura 21:



Fonte: Vilain (2002) citado por Hennrichs (2005).

Figura 21: Classes em Contexto.

Hennrichs (2005) identifica em Schwabe e Vilain (2002) a presença dos cartões de identificação como “ferramentas que o OOHDM propõe como auxílio na modelagem de uma aplicação” permitindo documentar todos aspectos de estrutura de acesso e contextos navegacionais.

Abaixo, alguns exemplos de cartões navegacionais:

Contexto: Professor por Área
Parâmetros: a:Área
Elementos: p:Professor where p associado_a a
Classe em contexto:
Ordenação: por p.nome, ascendente
Navegação interna: por índice (Idx Professores por Área(a))
Operações:
Usuários: alunos
Permissão: leitura
Comentários:

Estrutura de Acesso: Professores	
Parâmetros:	
Elementos: p:Professor	
Classe em contexto:	
Atributos:	Destino:
p.nome	Ctx Professor Alfabético (self)
p.grau	Ctx Professor por Grau (self.grau)
Ordenação: por p.nome, ascendente	
Restrições de Uso	
Usuários: alunos	Permissão: leitura
Comentários:	
Depende de: Nav_Professor	Influencia: ADV Professores

Fonte: Shwabe e Vilain (2002) citado por Hennrichs (2005).

Figura 22: Exemplos de Cartões de Identificação.

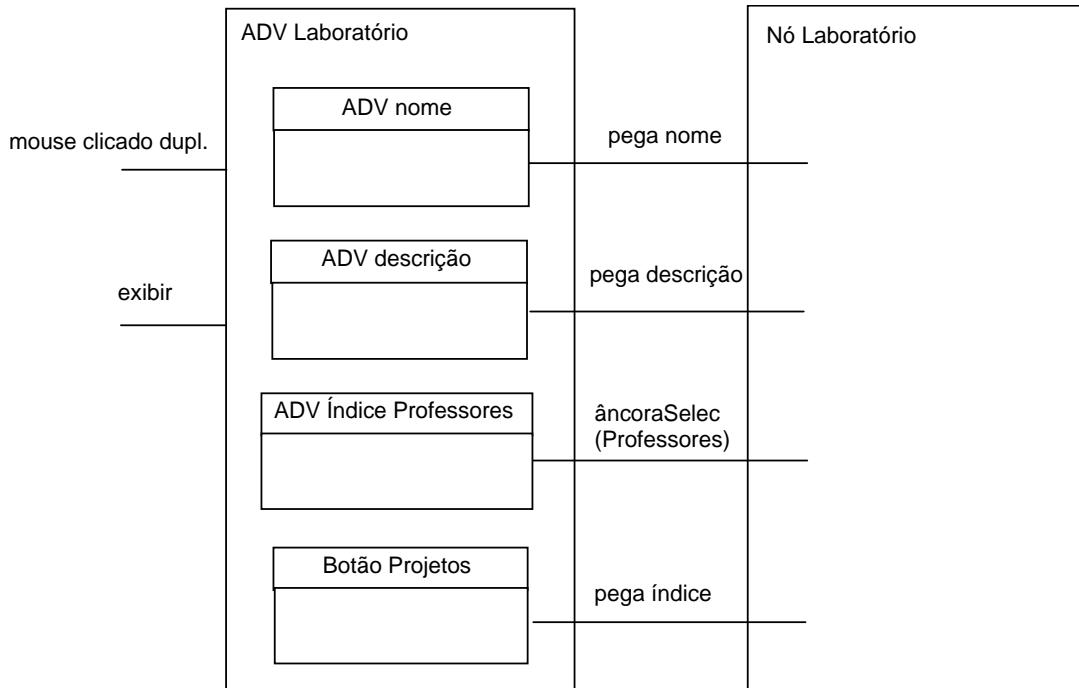
4.5 PROJETO DA INTERFACE ABSTRATA

Conforme Locatelli (2003, p.36), Schwabe e Vilain (2002, p. 74 - 77) e Schwabe, Pontes e Moura (1999), o projeto de interface abstrata tem como objetivo definir como serão distribuídos os objetos de interface (imagens, botões, textos, entre outros), suas propriedades e transformações, e também será responsável pela alta coesão e baixo acoplamento destes objetos. Para isto, o OOHDM utiliza o ADV (Abstract Data View) para especificar a aparência e interface dos objetos da aplicação. Além dos ADVs, existe outro objeto de interface chamado ADO (Abstract Data Objects), que, diferente dos ADVs, não possui a capacidade de reagir a eventos externos, servindo apenas para interagir com estruturas de dados e controle da aplicação.

Segundo Rossi (1997) citado por Hennrichs (2000, p. 57), o OOHDM utiliza “diagramas de configuração e ADVCharts para especificar o relacionamento estático e dinâmico, respectivamente, entre ADVs e ADOS”.

4.5.1 Diagramas de Configuração

Os diagramas de configuração possuem como objetivo: representar os relacionamentos entre os objetos de interface e os objetos navegacionais; incluir na aplicação, os objetos de interface ADVs especificando os eventos externos iniciados pelos usuários e manipulados pelos ADVs; especificar os relacionamentos estruturais entre os ADVs e os relacionamentos entre os ADVs (objetos de interface) e os ADOs (objetos de navegação).



Fonte: Shwabe e Vilain (2003, p. 76).

Figura 23: Diagrama de Configuração.

A figura 23 demonstra a comunicação entre o ADV Laboratório e o ADO Nó Laboratório, identificando que somente o ADV está reagindo as eventos externos, neste caso o clique do mouse (SCHWABE & VILAIN, 2003, p. 75).

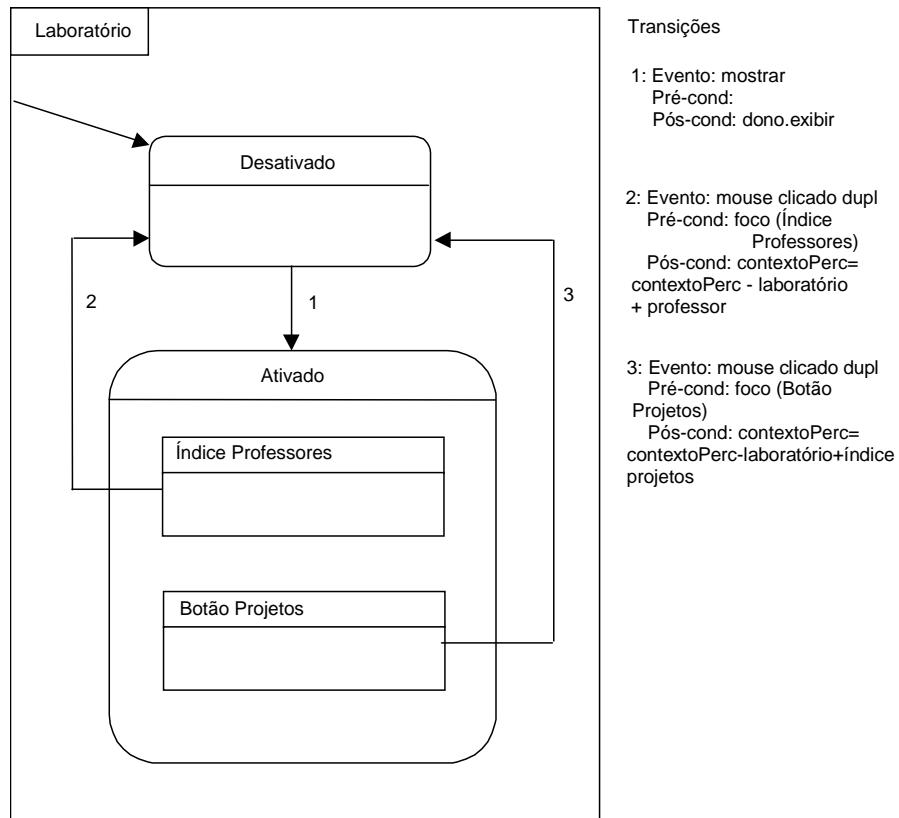
4.5.2 ADVcharts

Segundo Coelho (1995), citado por Hennrichs (2000), os ADVcharts objetivam ressaltar as transformações no nível da interface do usuário e seu impacto nos objetos navegacionais, apresentando as transformações ocorridas nos ADVs.

Cada ADVcharts pode possuir um ou mais ADVs, estados, atributos e transições, sendo que os estados dos ADVs são conectados por transações e alterados por eventos externos ou internos.

“Os ADVs são representados por retângulos, os estados por retângulos com bordas arredondadas e as transições por retas direcionadas” (SCHWABE & VILAIN, 2003, p. 76).

A figura abaixo demonstra um exemplo de um ADVchart de um Laboratório de Informática.



Fonte: Shwabe e Vilain (2002, p. 77).

Figura 24: ADVchart Laboratório.

4.6 IMPLEMENTAÇÃO

Esta é a fase final do OOHDM, e somente será realizada quando todas as fases anteriores estiverem concluídas. Consiste na tradução do projeto conceitual, navegacional e de interface para uma linguagem de implementação. Nesta fase define-se o ambiente em que o sistema vai ser implementado.

O OOHDM possui um ambiente de implementação para web chamado OOHDM-Web, no qual é permitida a automatização da criação dos nós, contextos e estruturas de acesso definidos na modelagem utilizada, bem como o OOHDM-ML desenvolvido com base no XML e o ambiente OOHDM-Java2, baseado na plataforma J2EE (HENNRICH, 2005).

Mesmo com ambientes de implementação já existentes, o OOHDM permite sua implementação em qualquer ambiente, ou seja, no ambiente que melhor convir a seu desenvolvedor, seja ele proprietário ou open source.

Segundo Hennrichs (2005), Rossi (1997) afirma que a implementação de um sistema elaborado pelo OOHDM se torna mais fácil se a linguagem tiver total suporte a orientação a objeto(OO). Porem o OOHDM permite a implementação tanto orientada a objetos como estruturada, e ainda, se necessário, ambas no mesmo projeto.

Durante esta fase também é realizado o mapeamento dos objetos apresentados no esquema conceitual para um modelo relacional. Obedecendo as seguintes regras (SHWABE & VILAIN, 2002):

- cada classe é mapeada como uma tabela;
- cada atributo da classe é definido como um atributo da tabela correspondente;
- caso haja um atributo multi-valorado (Ex.:Biografia:[Descrição+,Imagem]) é definida novas tabelas para estes atributos, com referência a tabela da classe conceitual;
- os relacionamentos com cardinalidade 1:N geram um campo de índice estrangeiro (FK – Foreign Key) na tabela postada no lado do N;
- os relacionamentos com cardinalidade N:N exigem a criação de uma tabela relacional entre as tabelas que representam as classes conceituais.

Antes da implementação é permitido a aplicação de Design Patterns (Padrões de Projeto) para a otimização e melhor solução das classes desenvolvidas no modelo conceitual.

5 ESCOTISMO

O estudo de caso deste trabalho tem como objetivo desenvolver uma WebApp que auxilie no gerenciamento de um grupo escoteiro, que é uma honrosa e trabalhosa tarefa, que segundo Baden-Powell (1995), trata-se da formação de jovens em cidadãos dignos e patriotas, com total aproveitamento de suas faculdades físicas, sociais, intelectuais, afetivas e espirituais, através do método e dos princípios escoteiros.

Conforme comentado na introdução deste trabalho, a escolha do tema, para aplicação da WebE e desenvolvimento de uma solução, ser o escotismo é referente ao fato deste movimento depender única e exclusivamente de atividades, de grande porte social, e baixos ou insignificantes custos e o grande conhecimento, prática e afinidade do autor do trabalho sobre esse assunto. Pois esse movimento se trata de uma fraternidade sem fins lucrativos que visa única e exclusivamente à educação não formal de jovens dos 07 aos 21 anos com o auxílio de adultos voluntários.

Muitas são as dificuldades gerenciais apresentadas por um grupo escoteiro, como:

- controlar o histórico e desenvolvimento individual de cada membro cadastrado como escoteiro ou escotista, devido a grande e crescente quantidade de informações, e a dificuldade de manter o currículo de cada componente (Escoteiro ou Escotista) atualizado.
- gerenciar as atividades existentes no escotismo, devido a alta quantidade e baixa documentação destas atividades, exigindo disponibilidade para sua busca e preparação, e havendo pouco tempo dos escotistas para elaboração destas.
- gerenciar o financeiro dos associados, permitindo visualização da validade dos registros dos jovens junto a UEB (União dos Escoteiros do Brasil), e controle das relações financeiras entre associados (pais) e o GE.
- controlar a agenda de atividades, evitando choques de datas e eventos entre o mesmo grupo e com o calendário oficial da UEB.

Porém, dentre todas estas necessidades, o problema de maior destaque se dá ao controle do histórico e desenvolvimento pessoal de cada escoteiro (lobinho, escoteiro, sênior e pioneiro) que é registrado sobre uma ficha intitulada Ficha 120, conforme será visto no próximo capítulo.

Tanto Shwabe e Rossi quanto Pressman e Ginige estruturam que antes da aplicação de uma engenharia, seja ela qual for, em um determinado problema, faz-se necessária uma visão geral deste ambiente para um melhor entendimento e então resolução das necessidades apresentadas por este caso real.

A presença deste capítulo “Escotismo” em um trabalho de processo de desenvolvimento de uma WebApp utilizando WebE e o Método OOHDM é necessário para permitir uma melhor integração, na visão do leitor, entre os problemas levantados neste ambiente (escotismo) e as soluções propostas pela WebE e resolvidas pelo OOHDM.

5.1 DEFINIÇÕES E CONCEITOS

Definição: “O escotismo é um movimento educacional para jovens, com a colaboração de adultos, voluntários, sem vínculos político-partidários, que valoriza a participação de pessoas de todas as origens sociais, raças e crenças, de acordo com o propósito, os princípios e o método escoteiro, concebidos pelo fundador, Baden Powell.” (P.O.R.²², 2001)

O escotismo surgiu da pioneira idéia de seu fundador Lord Robert Stephenson Smith Baden-Powell, hoje conhecido no mundo inteiro como Baden-Powell ou somente B-P (bipí).

As idéias de B-P eram únicas e originais para a época em que foram divulgadas. Essas idéias consistem na formação de cidadãos íntegros, através do auto-desenvolvimento. Os jovens, agindo em pequenas equipes formadas por quatro a oito integrantes, sob a liderança de um deles (o monitor) eleito pelos demais, assumiriam o compromisso de sua própria educação, sob a supervisão de um voluntário adulto (Chefe Escoteiro ou Escotista).

Formou-se assim o escotismo: um grupo de jovens, orientados por um ou mais adultos, realizando atividades diversas e interessantes, seguindo um método que valoriza o aprender fazendo, vida ao ar livre, um ambiente fraternal, fazendo com que a soma dessas experiências retornem como valores para suas personalidades, desenvolvendo suas potencialidades e o crescimento pessoal de cada um, como indivíduos felizes, participativos e úteis a sua comunidade (NAGY, 1985; HORN²³,2004).

²² P.O.R.: Princípios, organizações e regras. Manual estabelecido pela UEB, em sua última atualização 2001.

²³ Luiz César Horn: Integrante ativo da CNPJ (Comissão Nacional de Programa de Jovens) 2004. Responsável por instruir escotistas na aplicação do Programa de Jovens da UEB.

No Brasil o escotismo é identificado pela figura 25 e apresenta-se hoje como um dos países pioneiros no desenvolvimento do MACPRO (Método de Atualização e Criação Permanente do Programa de Jovens) disponível em Escoteiros(2005).



Fonte: P.O.R. (2001).

Figura 25: Emblema da UEB.

5.2 ESTRUTURA ORGANIZACIONAL NO ESCOTISMO

O movimento escoteiro está dividido em cinco categorias específicas de pessoas, também chamado **ramos**, que são classificadas conforme a idade do membro (P.O.R.,2001).

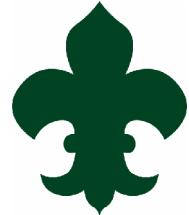
Dos 07 aos 10 anos o membro pertence ao **Ramo Lobinho** (Figura 26).



Fonte: P.O.R. (2001).

Figura 26: Símbolo do Lobismo.

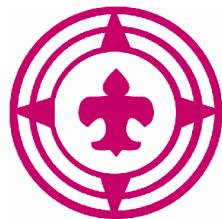
A partir dos 10 anos, até os 15 anos o membro passa a pertencer ao **Ramo Escoteiro** (Figura 27).



Fonte: P.O.R. (2001).

Figura 27: Símbolo do Ramo Escoteiro.

Quando chega aos 15 anos, até os 18 anos o membro pertence ao **Ramo Sênior** (Figura 27).



Fonte: P.O.R. (2001).

Figura 28: Símbolo do Ramo Sênior.

Entre seus 18 anos até seus 21 anos o membro passa a pertencer ao **Ramo Pioneiro** (Figura 29).



Fonte: P.O.R. (2001).

Figura 29: Símbolo do Ramo Escoteiro.

A partir dos 21 anos, todos membros passam a pertencer ao **Ramo Escotista** (Chefe Escoteiro) e/ou a Diretoria Executiva (Dirigentes).

5.2.1 O Grupo Escoteiro (GE)

O Grupo Escoteiro, conceituado no P.O.R. (2001), é uma organização local onde se aplica o escotismo aos jovens. Ele pode ser dividido em seções, sendo que cada seção é um grupo menor de pessoas, que irá abrigar um determinado número de membros de um ramo específico. As seções estão organizadas como alcatéias – para o ramo lobinho; tropa escoteira – para o ramo escoteiro; tropa sênior – para o ramo sênior; e clã pioneiro – para o ramo pioneiro. Cada seção possui um responsável, nomeado chefe de seção. O grupo pode possuir nenhuma, uma ou mais seções de cada ramo, tendo em vista que o GE necessita ter ao menos uma seção ativa.

O GE também possui, além das seções, três outros órgãos, estabelecidos pelo Estatuto da UEB (2004):

- Assembleia do Grupo: responsável pela eleição da diretoria do grupo, bem como pela escolha da Comissão fiscal. Ela é responsável também pela tomada de decisão da “alienação ou a oneração” dos bens imóveis do grupo através do estatuto e também supervisionar, através de relatórios, as atividades da diretoria, comissão fiscal e das seções do GE. É composta por três membros da diretoria do grupo, pelos escotistas, pioneiros, participantes contribuintes da UEB e vinculados ao GE e pelos membros juvenis, caso o regulamento do GE permita. Possui um presidente e um secretário, eleitos entre eles, em cada reunião;
- Diretoria do Grupo: responsável pelo bom desenvolvimento do GE, fazendo cumprir o estatuto do grupo, o P.O.R. e regulamentos da UEB. É composta por, no mínimo, três elementos, eleitos pela assembléia do grupo, sendo que, um deles seja o Diretor Presidente, responsável pela coordenação, direção e representação do grupo. A diretoria também é responsável por assegurar a continuidade e o desenvolvimento do GE, como também manter em dia o registro de todos os membros pertinentes a ele;
- Comissão Fiscal do Grupo: responsável pela fiscalização e fiscalização da gestão financeira e patrimonial do grupo. A comissão é composta por três elementos que possuem mandato de dois anos. Ela auditora relatórios de atividades dos escoteiros e os balancetes semestrais do GE.

Os demais voluntários presentes no GE são nomeados assistentes de seção, sendo suas funções de responsabilidade do chefe de cada seção.

Cada membro escotista de um GE possui um **Assessor Pessoal de Formação**. Essa pessoa será responsável pela formação deste membro dentro do escotismo. (P.O.R.,2001)

Há nas seção do GE uma sub-divisão, aplicada somente nas alcatéias e tropas, na qual os jovens são divididos em pequenos grupos de quatro a oito integrantes, denominados matilha para o ramo lobinho e patrulha para os ramos escoteiro e sênior. A vida em equipe é visualizada nesta divisão, fazendo com que os jovens vivam em uma pequena sociedade particular, estimulando-os ao companheirismo e espírito de equipe, mencionado no método escoteiro. Essa sub-divisão também recebe nomenclaturas conforme o ramo a que pertence. Esses nomes são: cores (branca, preta, cinza, dentre outras) para o ramo lobinho, animais (panda, pantera, águia, tigre, dentre outras) para o ramo escoteiro e locais indígenas ou constelações (Charrua, Tamoio, Fênix, entre outras) para o ramo sênior. Conforme já mencionado, cada patrulha possui um integrante responsável, geralmente, eleito pela patrulha.

5.3 FICHAS E CERTIFICADOS

Um conjunto de documentos se faz necessário para o controle e bom funcionamento de um grupo escoteiro e para a correta prática do escotismo. Dentre estes documentos, serão citados alguns de suma importância e utilidade (HORN,2003).

5.3.1 Fichas REME

As Fichas REME (Rede de elaboração de material educativo) são ferramentas que foram desenvolvidas por integrantes do movimento escoteiro e padronizadas pela Interamerican Scout Office (Oficina Internacional Americana), com o objetivo de facilitar o planejamento, desenvolvimento e a avaliação das atividades aplicadas aos jovens de acordo com o método e os objetivos traçados pelo escotismo.

5.3.2 Ficha 120 e 121

A Ficha 120 é o documento oficial de registro do membro juvenil junto a UEB. Tem como objetivo registrar informações pessoais e o histórico de cada jovem. Esta ficha é composta por seis áreas de preenchimento: dados do GE, dados pessoais do escoteiro, datas das

conquistas e progressões, especialidades com data e nível e histórico da vida escoteira do membro juvenil (ESCOTEIROS, 2003).

A Ficha 121 possui o mesmo papel da Ficha 120, porém aplicada e adaptada aos membros escotistas (também conhecidos como CHEFES) do GE conforme o site oficial da UEB²⁴ (2004).

Fonte: Página <http://www.escoteiros.org.br> (2004).

Figura 30: Modelo de Ficha 120.

²⁴ <<http://www.escoteiros.org.br>>

5.3.3 Certificados

Os certificados são responsáveis por documentar datas e marcos na vida escoteira de um membro. Eles possuem um padrão de *layout* estabelecido pelo P.O.R. (2001) e têm as seguintes responsabilidades:

Certificado de Promessa: certificado responsável por documentar a data e local de promessa do membro, no movimento escoteiro.

Certificado de Especialidades: certificado responsável por documentar a data, local e nível de especialização adquirida pelo membro juvenil em determinada área.

Certificado de Progressão: certificado responsável por documentar a data, local e nível de progressão adquirido pelo membro juvenil em seu ramo.

Certificado de Participação em eventos: certificado responsável por documentar data e local de participação do membro escoteiro em determinado evento decorrente do movimento escoteiro.

Certificado de Atividades: certificado responsável por documentar o tempo de participação, em anos, do membro no movimento escoteiro.

Certificado de Condecorações: certificado responsável por documentar data, local e homenagem concedida ao cidadão que tenha prestado bons serviços ou atos de heroísmo ou bravura no movimento escoteiro.

Certificado de Cursos: certificado responsável por documentar data, local e nível de curso do membro escotista.

Certificado de Nomeação: certificado responsável por documentar a nomeação (área de atuação) do membro escotista.

Há vários outros documentos e certificados utilizados para a administração de um GE. Para visualização e entendimento de todos documentos é recomendada a leitura do já citado site da UEB e do P.O.R. (2001).

5.4 GE XIMBANGUE



Figura 31: Logotipo do GE Ximbangue.

Situado em Chapecó, homenageia em seu nome as comunidades indígenas desta região, bem como, um dos principais índios destas comunidades, conforme figura 31.

Completou 20 anos no dia 04 de Agosto de 2004. Possui cerca de 110 membros, sendo eles:

- 20 escotistas;
- 03 dirigentes;
- 87 jovens.

O GE Ximbangue realiza suas atividades todos os sábados das 13:30 às 16:30 horas e sua sede própria localizada no Parque Verdão de Chapecó.

Dentre os grupos escoteiro de Santa Catarina apresenta-se entre os dez maiores e um dos mais bem preparados para a aplicação do escotismo segundo a UEB.

Sua estrutura organizacional divide-se atualmente entre:

- 01 alcatéia mista;
- 01 tropa escoteira mista;
- 01 tropa sênior mista;
- 01 clã pioneiro misto.

6 ENGENHARIA DE UMA WEBAPP PARA UM GRUPO ESCOTEIRO

UTILIZANDO O MÉTODO OOHDM

O desenvolvimento da WebApp para o gerenciamento de um Grupo Escoteiro (GE), mais especificamente para o gerenciamento das atividades dos membros e controle de seu histórico será guiado pela seqüência de atividades propostas pelo método OOHDM e descrito no capítulo 4.

Para cada atividade realizada no método OOHDM será dado ênfase nas respectivas atividades que devem ser realizadas pelo modelo apresentado por Pressman (2002), visualizado na Figura 3 e discorrido no capítulo 3 que faz referência sobre Engenharia Web. A interação entre as duas seqüências de atividades (modelo de Presmann e OOHDM), pode ser visualizada na figura abaixo:

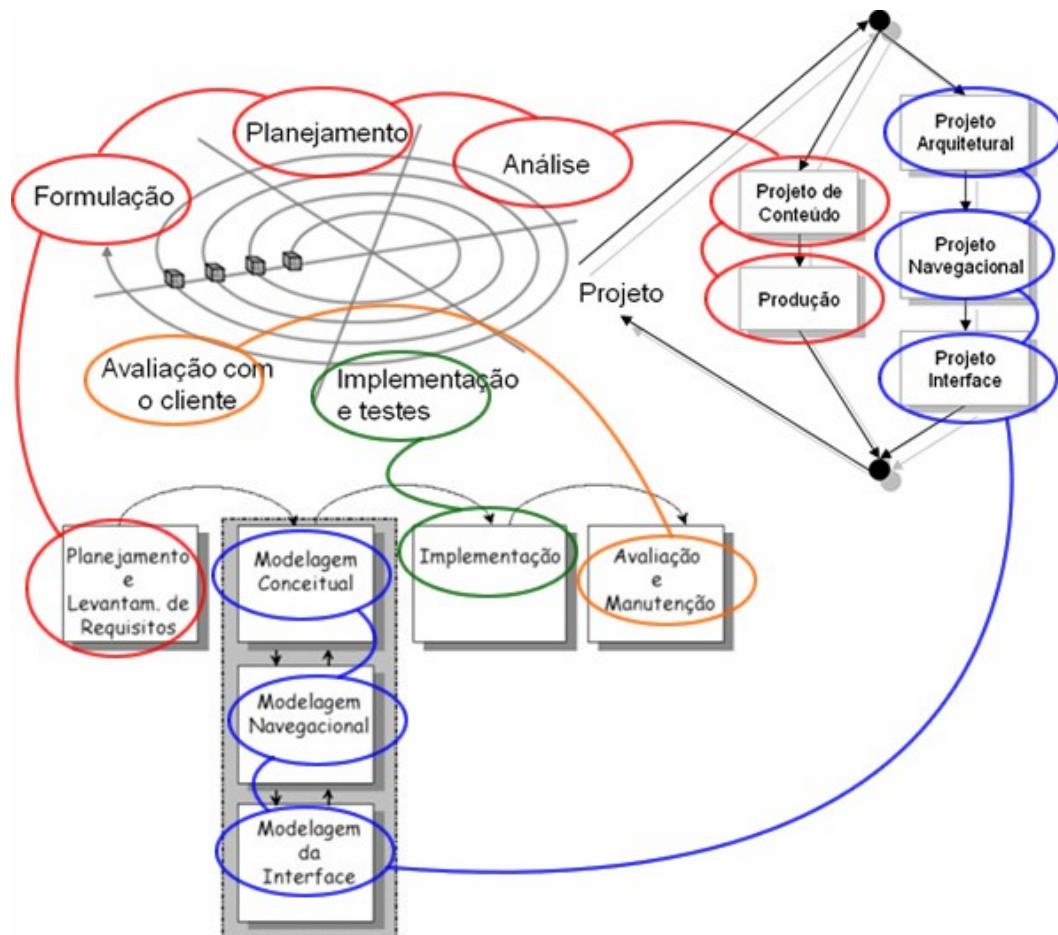


Figura 32: Estrutura de Desenvolvimento da WebApp.

A partir de agora para cada uma das fases do OOHDIM será abordado todos os procedimentos práticos aplicados neste trabalho, com suas necessidades, fundamentos e resultados. Referenciando-se em cada etapa as fases da WebE abordadas, conforme sugerido na Figura 32.

6.1 LEVANTAMENTO DE REQUISITOS

Conforme a figura 32, os campos destacados em vermelho identificam que a fase do **planejamento e levantamento de requisitos** do OOHDIM engloba todas as atividades referentes ao levantamento de informações para o desenvolvimento da WebApp proposto por Pressmann: **formulação, planejamento e análise**, bem como o **projeto de conteúdo e produção** responsáveis pelos conteúdos informacionais da WebApp.

As questões apresentadas na formulação do projeto do tipo: “*porquê?*”, “*para quem?*” e “*como?*” são esclarecidas na identificação dos atores e tarefas. Ao identificar esses dois itens as metas de informação e aplicativo estarão completamente definidas e esquematizadas.

Nesta etapa, o OOHDIM permite maior contato com os usuários do sistema, permitindo que os riscos levantados pela análise e as permissões e rotinas de atividades da aplicação sejam adequadas e aprovadas por eles. Isto é possível através das UIDs que são apresentadas, aos usuários do sistema, através de uma linguagem de fácil entendimento.

No estudo de caso para o desenvolvimento de uma WebApp para o gerenciamento de um grupo escoteiro será focada a atividade de preenchimento e gerenciamento das, agora denominadas somente, “Ficha 120”.

Conforme visualizada na figura 30 do capítulo cinco, esta ficha é responsáveis pelo arquivamento e registro das atividades realizadas pelos membros juvenis no escotismo. Dentre elas:

- Os dados gerais do grupo escoteiro;
- Os dados pessoais do escoteiro;
- Datas de conquistas e progressão de cada escoteiro;
- Relação de especialidades realizadas pelos escoteiros;

- A data e relação da vida escoteira de cada escoteiro.

Entende-se que cada atividade, especialidade, conquista e progressão realizada por um escoteiro é documentada de forma manual sobre essa ficha pelo chefe encarregado. Levando em consideração que em cada seção existente em um grupo escoteiro, há, em média, de vinte a trinta e cinco escoteiros. Então para cada atividade realizada por uma seção, por exemplo um acampamento, deverá ser documentada manualmente, e de forma idêntica, em cada ficha de cada escoteiro presente neste acampamento.

A manutenção desta Ficha 120 é uma, senão, a mais trabalhosa e redundante tarefa necessária para o bom funcionamento do escotismo, conforme levantamento realizado sobre os chefes presentes no GE Ximbangue em uma reunião de chefes e dirigentes datada em 26 de fevereiro deste ano de 2005.

6.1.1 Identificação de Atores e Tarefas

Após a definição do escopo, na entrevista entre os chefes e dirigentes do GE Ximbangue, iniciou-se a primeira etapa da fase de levantamento de requisitos do OOHDm: **a identificação de atores e tarefas.**

A manipulação e gerenciamento da Ficha 120 será realizada por três atores:

- Dirigentes
- Chefes (escotistas)
- Escoteiros (membros juvenis)

Para cada ator foram levantadas suas possíveis tarefas relacionadas ao escopo da WebApp:

1. Ator: Dirigente

- 1.1. Alterar informações institucionais do GE;
- 1.2. Cadastrar uma nova seção no GE;
- 1.3. Cadastrar um novo chefe em uma seção;
- 1.4. Ver escoteiros por seção;
- 1.5. Ver ficha 120 de um escoteiro;
- 1.6. Ver chefes por seção;
- 1.7. Reativar um membro no GE;
- 1.8. Trocar um chefe de seção;
- 1.9. Cadastrar um novo dirigente no GE;
- 1.10. Ver dirigentes presentes no GE;
- 1.11. Ver detalhes de um membro;
- 1.12. Alterar alguns campos dos seus dados cadastrais.

2. Ator: Chefe

- 2.1. Ver todos os escoteiros de sua seção;
- 2.2. Ver os chefes pertencentes a sua seção;
- 2.3. Cadastrar um novo escoteiro em sua seção;
- 2.4. Transferir um escoteiro de sua seção para outra;
- 2.5. Desativar um escoteiro de sua seção;

- 2.6. Registrar a data de uma conquista realizada por um escoteiro de sua seção;
- 2.7. Registrar a data de uma progressão atingida por um escoteiro de sua seção;
- 2.8. Registrar a data e nível de uma especialidade realizada por um escoteiro de sua seção;
- 2.9. Cadastrar uma atividade à sua seção;
- 2.10. Ver e imprimir a ficha 120 de um escoteiro;
- 2.11. Ver detalhes de um membro
- 2.12. Ver os dirigentes do seu GE
- 2.13. Alterar alguns campos dos seus dados cadastrais.

3. Ator: Escoteiro

- 3.1. Alterar alguns campos dos seus dados cadastrais;
- 3.2. Ver todos os chefes de sua seção;
- 3.3. Ver todos os escoteiros de sua seção;
- 3.4. Ver detalhes de um membro;
- 3.5. Ver os dirigentes do seu GE;
- 3.6. Ver e imprimir sua Ficha 120;

Observa-se que após a identificação dos atores e tarefas ficam claras as questões:

- **Qual o objetivo principal da aplicação a ser desenvolvida?**

Auxiliar na manutenção e gerenciamento das informações dos membros juvenis de um GE (Escopo definido em reunião).

- **Qual será a utilidade desta aplicação?**

Relação de tarefas apresentadas a cada ator da aplicação.

- **Quais serão os usuários desta aplicação?**

Os atores identificados: dirigentes, escoteiros e escotistas.

6.1.2 Especificação de Cenários

Para cada tarefa identificada na fase anterior será criado um cenário. Este cenário será responsável pela contextualização da tarefa e a sua forma de resolução. Neste momento são apresentados os dados que serão trocados entre os usuários (Atores) e a aplicação.

O modelo de cenário apresentado por Shwabe e Vilain (2002) é este:

Cenário: Código do cenário (Atores) – Tarefa relacionada

Contexto: O motivo de existir esta tarefa

Objetivo: Objetivo a ser atingido por esta tarefa

Ações: Roteiro superficial da seqüência de atividades

As palavras destacadas em negrito servem de índice para as informações contidas em cada cenário (HENNRICHS, 2005; SHWABE & VILAIN, 2002).

Em **Cenário:** é feita a catalogação de cada cenário, dando-lhe um código único identificado como código do cenário. É permitido destacar o ator que está ativo neste cenário entre parênteses “()”, então descreve-se a tarefa relacionada a este cenário.

No **Contexto:** é apresentada uma situação real para ocorrência deste cenário, ou seja, um exemplo real justificando e apresentando a necessidade desta tarefa para a aplicação. O contexto auxilia ao desenvolvedor confirmar se realmente identificou qual a necessidade da tarefa em questão. Possibilitando com isso, uma concordância ou discordância da atividade indicada em uma apresentação deste contexto ao futuro usuário do sistema.

O **Objetivo:** apresenta de forma clara e direta o principal objetivo da tarefa em questão, permitindo, se o objetivo está se tornando complexo, redefinir a tarefa em tarefas mais simples e diretas, até que tornem o objetivo direto e claro.

As **Ações:** devem apresentar uma seqüência breve e direta de como a tarefa vai ser executada para chegar ao objetivo desejado, sem levar em consideração análises de interface ou procedimentos técnico da aplicação.

A seguir é apresentado um exemplo de cenário que foi abstraído de uma tarefa já coletada anteriormente com os atores. O conjunto completo destes cenários está apresentado no Anexo I deste trabalho.

Tabela 3: Exemplos de cenários da WebApp para um GE

Cenário: C1.2 (Dirigente) – Cadastrar uma nova seção no GE.
Contexto: Sou dirigente de um grupo escoteiro e acabamos de inaugurar uma nova alcatéia devido a grande procura de crianças pelo movimento escoteiro e a disponibilidade de chefes no GE. Precisamos então criar essa nova seção para que possam ser transferidos os devidos chefes para esta, bem como, os novos membros juvenis.
Objetivo: Inserir uma nova seção no GE.
Ações: Eu acesso a área de gerenciamento de seções e escolho a opção de criação de nova seção. Informo o nome e o ramo que esta nova seção possuirá, bem como, a data de criação e algumas observações, caso necessário. Confirmo os dados que acabei de inserir e o sistema realiza o cadastro desta nova seção no GE.

Após a diagramação e correção dos cenários, é necessário a classificação e agrupamento destes para uma melhor generalização, permitindo futuras especializações, propostas em ambientes orientados a objeto.

6.1.3 Especificação dos Use Cases

A etapa de especificação dos use cases terá por objetivo generalizar os cenários já existentes com o intuito de melhor apresentar a seqüência de atividades para a realização das tarefas desejadas.

Shwabe e Vilain (2002) apresentam o seguinte modelo para relatar os use cases:

Use Case: Descrição do use case

Cenários: C1(Ator)....

Descrição:

1. Atividade 01
2. Atividade 02
3.

As palavras destacadas em negrito servem de índice para as informações contidas em cada use case (HENNRICHS, 2005; SHWABE & VILAIN, 2002).

Em **Use Case**: é dado um nome para cada use case de forma a catalogá-lo. Este nome expressa diretamente, como o objetivo do cenário, a atividade que este use case se propõe a realizar.

Em **Cenários**: são relacionados os códigos de todos os cenários que foram agrupados para este use case.

A **Descrição**: relata numericamente, passo a passo, toda interação realizada entre o usuário e a aplicação, dando uma seqüência para realização das atividades solicitadas nos cenários.

Em uma analogia com a orientação a objetos, podemos identificar os cenários como objetos e os use cases como as classes que o originaram (RUMBAUGH, 1994, apud SHWABE & VILAIN, 2002).

A seguir será apresentado um use cases, bem como sua descrição, identificado agrupando-se os cenários já relatados anteriormente. O conjunto completo destes use cases está no Anexo I deste trabalho.

Tabela 4: Exemplo de um use cases da WebApp para um GE

Use Case 4:	Ver relação de escoteiros por seção
Cenários:	C1.4/ C2.5/ C2.1/ C3.3
Descrição:	
<ol style="list-style-type: none"> 1. O sistema apresenta o nome e ramo das seções presentes no grupo atualmente. 2. O usuário seleciona uma seção. 3. O sistema apresenta a relação de escoteiros presentes atualmente na seção com seu nome, registro na UEB, seção e ramo. 4. O usuário pode imprimir essa relação. 5. Caso o usuário seja um chefe ou dirigente ele pode ainda visualizar a ficha 120 do escoteiro, transferi-lo para outra seção ou desativá-lo, selecionando-o pelo nome. 6. Senão pode ver mais informações sobre o escoteiro selecionado (Ver detalhes sobre um membro). 	

O use case apresentado acima assume como nome de identificação “Ver relação de escoteiros por seção”. Observa-se que este nome indica de forma direta o objetivo a ser alcançado por este use case.

Após seu nome são apresentados os códigos dos cenários que compõem este use case:

- C1.4: (Dirigente) – Ver escoteiros por seção.
- C2.5: (Chefe) – Desativar um escoteiro da minha seção.
- C2.1: (Chefe) – Ver todos os escoteiros de sua seção.
- C3.3: (Escoteiro) – Ver todos os escoteiros de sua seção.

Em sua descrição é apresentados seis passos genéricos que possibilitam a realização dos cenários destacados. Sem levar em consideração questões de funcionamento interno da aplicação.

6.1.4 Especificação dos UIDs

A especificação dos UIDs conforme vista no capítulo 4, visa uma representação gráfica dos *Use Cases* com o objetivo de apresentar as interações entre o usuário e o sistema sem detalhes referentes a interface com o usuário. Em resumo, é uma forma gráfica e de fácil entendimento, para se esquematizar a interação dos use cases, e que tem por objetivo, obter maiores detalhes e confirmação de definições em futuras entrevistas com os usuários.

A Figura 33 apresenta um exemplo de UID identificado a partir dos use cases gerados. O conjunto completo destes UIDs estão presentes no Anexo I deste trabalho.

Exemplos de UID da WebApp para um GE:

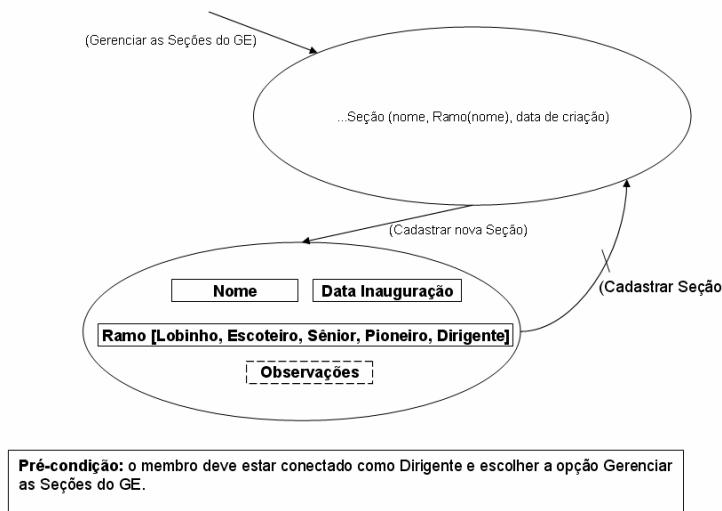


Figura 33: UID 02 – Cadastrar Seção

O UID inicia na seta que aponta para o balão superior, através de uma execução chamada (Gerenciar as Seções do GE), que pode vir a ser um link, opção de menu, um botão, entre outros.

A primeira interação (balão de cima) onde está o texto "...Seção (nome, Ramo(nome), data de criação)" apresenta uma relação, devido a reticências "...", das seções já presentes na aplicação com os seus dados especificados: nome da seção "nome", nome do ramo ao qual pertence "Ramo (nome)", e sua data de criação "data de criação".

Haverá ainda nesta interação uma opção identificada como “Cadastrar nova Seção”. Quando esta opção for selecionada será apresentada a nova interação solicitando para o usuário os campos contidos dentro das caixas de texto como: “Nome”, “Data Inauguração”, “Ramo [Lobinho, Escoteiro, Sênior, Pioneiro, Dirigente]” (os colchetes “[]” indicam que o usuário poderá fazer uma escolha entre os campos apresentados), e “Observações” (este campo, por estar com sua borda pontilhada, se torna opcional).

Nesta interação o usuário ainda visualizará uma opção chamada “Cadastrar Seção” que permitira o cadastro da seção desejada e a volta para visualização das seções existentes no sistema. O pequeno traço no relacionamento “(Cadastrar seção)” indica que esta opção não tem retorno após ser selecionada.

Em alguns casos haverá um campo de texto abaixo do UID, contornado por uma caixa com borda escura indicando uma “Pré-condição” ou uma “Pós-condição” para ocorrência das interações do UID. Neste caso a “Pré-condição” indica que o usuário do sistema deve estar identificado como um Dirigente para poder acessar a opção de “Gerenciar as seções do GE”.

6.1.5 Validação dos Use Cases e UIDs

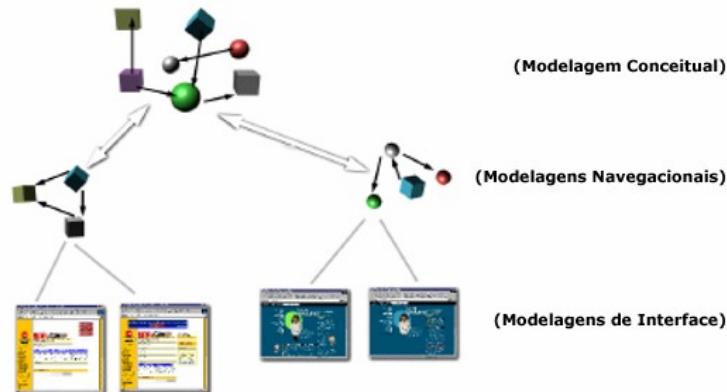
Após a confeção de todos os use cases e UIDs necessários ao escopo da aplicação é realizada uma nova entrevista entre os usuários da aplicação e os projetistas da mesma (VILAIN & SHWABE, 2002), para validação dos mesmos.

Esta atividade, no caso da WebApp do GE Ximbangue foi sendo realizada a medida que estes documentos foram sendo confeccionados, evitando assim erros no foco da aplicação, bem como, uma maior agilidade na realização destas etapas.

6.2 AS MODELAGENS

As fases de modelagem conceitual, modelagem navegacional e modelagem da interface do OOHDM, destacados em azul, englobam as três fases correspondentes na fase de projeto da engenharia web proposta por Pressman, que são: projeto arquitetural, projeto navegacional e projeto de interface. As fases denotadas em azul na Figura 32.

Neste momento, a análise de requisitos cumpriu sua tarefa de levantar subsídios para desenvolver as modelagens necessárias para a aplicação. As modelagens propostas pelo OOHDM, segundo Shwabe e Rossi (1998), seguem a hierarquia, conforme apresenta a figura abaixo:



Fonte: Rossi e Shwabe (1998)

Figura 34: Modelagens OOHDM

A partir de um modelo conceitual, são criados visões da aplicação, de acordo com os tipos de atores-usuários presentes na aplicação, estabelecendo-se as modelagens navegacionais. Com as modelagens navegacionais definidas, são projetadas as interfaces abstratas que serão responsáveis pela interação visual do usuário com a aplicação.

Para o GE Ximbangue, a WebApp de gerenciamento de Fichas 120 tem previsão de possuir uma única modelagem conceitual, sem sub-sistemas. Esta WebApp possuirá também três modelos navegacionais, um para cada ator identificado (Dirigente, Escoteiro e Chefe).

Cada um dos modelos navegacionais desenvolvidos requererá tantas interfaces forem necessárias, de acordo com o número de contextos existentes, para que cada ator tenha possibilidade de executar todas suas tarefas solicitadas no levantamento de requisitos.

6.3 MODELAGEM CONCEITUAL

Com toda documentação realizada na fase de levantamento de requisitos, iniciou-se o estudo sobre os UIDs a fim de desenvolver um diagrama conceitual para a WebApp.

O objetivo desta fase é gerar um diagrama navegacional, o que permitirá, para a engenharia de software, a visualização da arquitetura comportamental e referencial entre todas as atividades do sistema.

A cada UID analisado serão extraídas todas as possíveis classes, sem levar em conta a duplicidade ou outras regras de orientação a objetos (Classificação, agregação, generalização e espacialização).

As seguintes candidatas a classe foram encontradas após a análise dos UIDs:

- Seção e Ramo do UID 02;
- Seção, Ramo e Escoteiro do UID 04;
- GE, Escoteiro, Conquista, Progressão, Especialidade, Atividade e Ramo do UID 05;
- Seção, Ramo e Membro do UID 06;
- Atividade, Tipo, Local, Ramo e Escoteiro do UID 09
- Membro do UID 10;

Após encontrar todas possíveis classes, forma eliminadas as duplicidades, restando apenas:

- Seção
- Ramo
- Escoteiro

- GE
- Conquista
- Progressão
- Especialidades
- Atividades
- Membro
- Tipo
- Local

Com as possíveis classes identificadas foram atribuídos os possíveis atributos a cada uma delas, neste momento, já foram excluídos os atributos em duplicidade.

- Seção: nome, data de criação, observação.
- Ramo: nome.
- Escoteiro: nome, registro UEB, endereço, bairro, cidade, estado, CEP, telefone, e-mail?, celular?, data de nascimento, naturalidade, religião, nome do pai, profissão do pai, telefone do pai, nome da mãe, telefone da mãe, profissão da mãe e foto.
- GE: nome, região, numeral, estado, setor, site e e-mail.
- Conquista: nome e data.
- Progressão: nome e data.
- Especialidades: nome, data, nível e tarefas.
- Atividades: data e observações.
- Membro: nome, registro UEB, endereço, bairro, cidade, estado, CEP, telefone, e-mail?, celular?, data de nascimento, naturalidade, religião, nome do pai,

profissão do pai, telefone do pai, nome da mãe, telefone da mãe, profissão da mãe e foto.

- Tipo: nome.
- Local: nome.

A partir deste momento, são aplicadas as regras de classificação-agregação e especialização-generalização definidas na orientação a objetos, fazendo-se útil a relação de atores e tarefas para melhor definir os relacionamentos.

As operações e perspectivas também são obtidas mediante análise técnica do projetista perante as necessidades da aplicação apresentadas tanto na identificação dos atores e tarefas, como nos UIDs desenvolvidos.

Obtém-se então o modelo conceitual da WebApp (Figura 35).

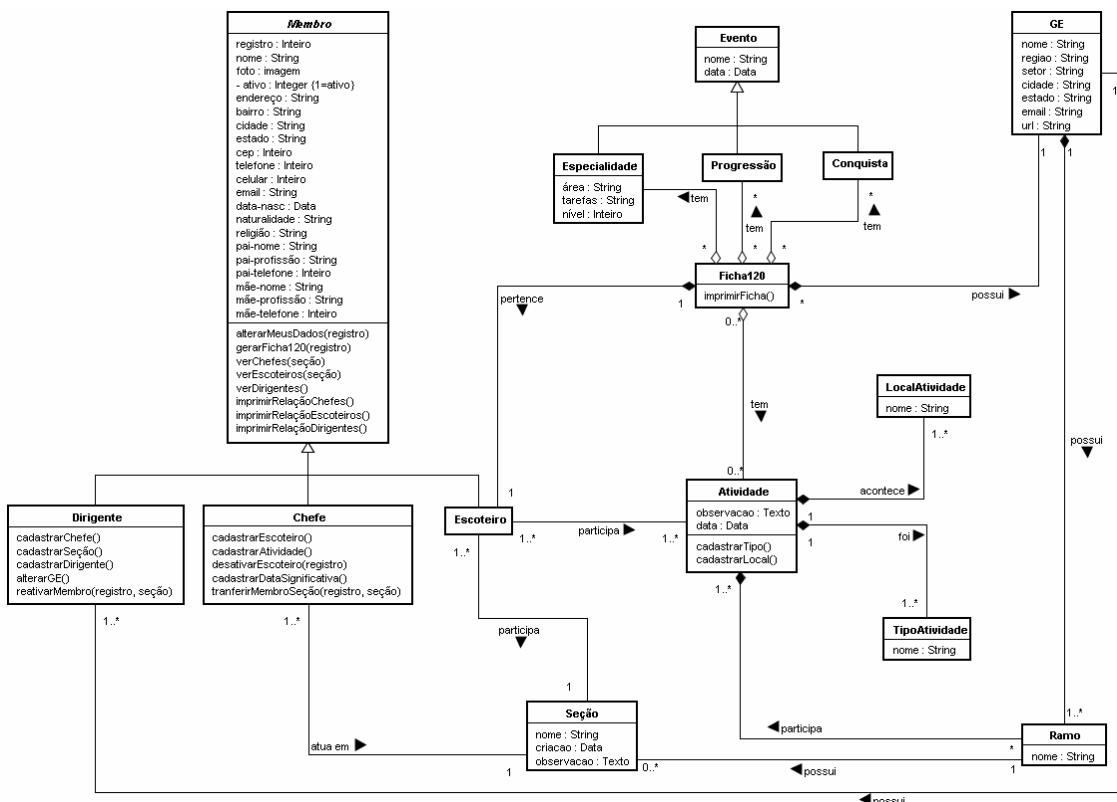


Figura 35: Modelo Conceitual da WebApp do GE Ximbangue

6.4 MODELAGEM NAVEGACIONAL

Definido o modelo conceitual, parte-se para a análise da modelagem navegacional, responsável pela visão navegacional de cada usuário perante a aplicação.

Esta fase objetiva fundamentar o relacionamento entre todas as atividades do sistema, solicitadas na fase de projeto navegacional da engenharia web, permitindo também o gerenciamento de permissões de usuários e o mapeamento navegacional para um usuário chegar a um resultado (tarefa solicitada no levantamento de requisitos para o usuário).

Nesta fase, retoma-se os perfis de atores identificados na análise de requisitos, bem como as tarefas que estes podem executar com o objetivo de auxiliar a melhor forma de apresentar os contextos navegacionais.

6.4.1 Esquema de Contextos

Cada contexto navegacional da aplicação baseou-se nas cinco diretrizes propostas por Shwabe (2003)²⁵ utilizando as UIDs já aprovados pelos usuários do sistema.

Cada diretriz consistem em identificar padrões nos UID, mapeando-os para os contextos navegacionais, das seguintes formas:

- **1^a Diretriz:** Mapeia-se um conjunto de estruturas para uma estrutura de acesso quando a tarefa pertinente ao UID requer a seleção de um, entre muitos elementos iguais, em um conjunto de elementos;
- **2^a Diretriz:** Mapeia-se um conjunto de estruturas para um contexto de navegação a partir do momento em que a tarefa pertinente ao UID solicita informações de um elemento em específico do conjunto;
- **3^a Diretriz:** igualmente como na 2^a diretriz, mapeia-se uma estrutura singular para um contexto de navegação a partir do momento em que a tarefa pertinente

²⁵ Apresentações de PowerPoint disponíveis em <http://www.telemidia.puc-rio/oohdm/Slides_Curso/>. Acessadas em outubro de 2004.

ao UID solicitar que as informações de um elemento, correspondentes a uma estrutura, seja acessada pelo usuário;

- **4^a Diretriz:** consiste em montar um contexto de navegação para retornar dinamicamente informações solicitadas por um usuário, por exemplo a busca por uma matéria através de uma palavra chave dentro de um contexto de Matérias por pesquisa;
- **5^a Diretriz:** mapeia-se uma estrutura singular para um contexto de navegação quando a tarefa pertinente ao UID solicitar diretamente um resultado de valores referente a uma entrada do usuário.

Aplicando-se estas diretrizes, bem como, levando em consideração as tarefas pertinente a cada ator existente na aplicação, chegou-se a definição de três contextos navegacionais, um para cada usuário da aplicação:

- Contexto Navegacional Ator Dirigente,
- Contexto Navegacional Ator Chefe, e
- Contexto Navegacional Ator Escoteiro.

O primeiro contexto definido foi o contexto navegaciona do Dirigente, conforme a figura abaixo:

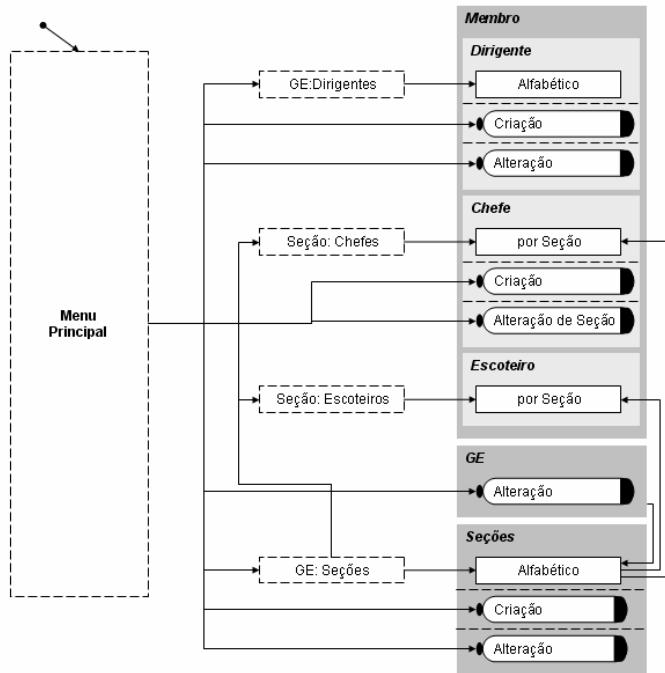


Figura 36: Contexto Navegacional Ator Dirigente

Este contexto permite visualizar que um usuário, identificado como dirigente, pode:

- acessar a relação de dirigente em ordem alfabética, a relação de chefes por seção, a relação de escoteiros por seção;
- acessar a relação alfabética de seções presentes no GE, bem como, os escoteiros e chefes pertinentes a essa seção;
- gerar, dinamicamente, novos dirigentes (Dirigente Criação);
- alterar os dados de um dirigente, neste caso, ele mesmo (Dirigente Alteração);
- gerar novos chefes (Chefs Criação) e transferir um chefe de seção (Chefe Alteração de Seção);
- alterar as informações de um GE (GE Alteração) e criar/alterar as seções do GE (Seção Criação/Seção Alteração).

O segundo contexto definido foi o contexto navegaciona do Chefe, confore a figura abaixo:

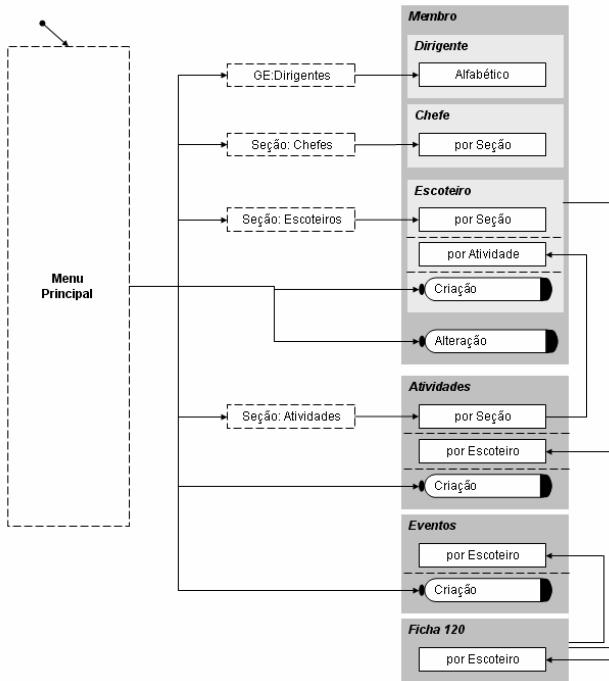


Figura 37: Contexto Navegacional Ator Chefe.

Este contexto demonstra que um usuário, identificado como chefe, ao acessar o sistema pode:

- ver a relação de dirigente em ordem alfabética e a relação de chefes por seção;
- a relação de escoteiros por seção, como também, os eventos, as atividades e a Ficha 120 de cada escoteiro;
- a relação alfabética de seções presentes no GE,
- a relação de atividades realizadas por cada seção, bem como, os escoteiros que participaram destas atividades;
- pode ainda cadastrar uma nova atividade a sua seção (Atividades Criação), cadastrar um novo evento a um escoteiro (Evento Criação) e alterar seus dados pessoais (Chefe Alteração).

O terceiro contexto definido foi o contexto navegaciona do Escoteiro, conforme a figura abaixo:

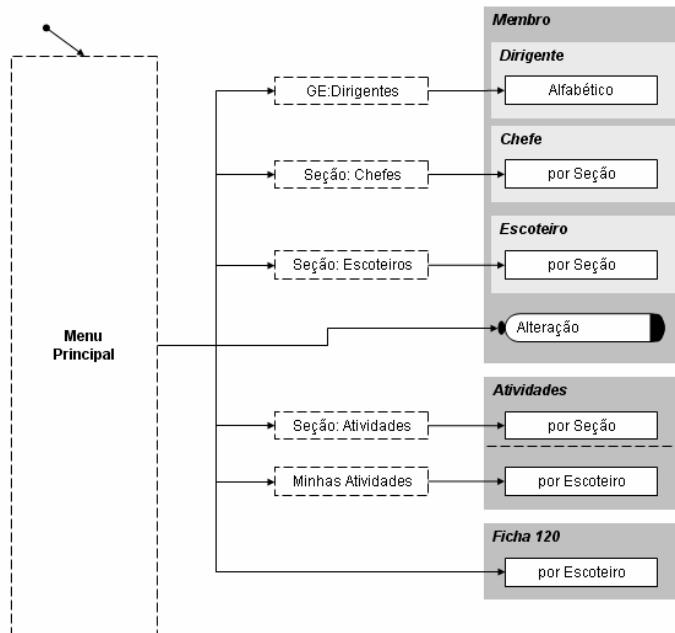


Figura 38: Contexto Navegacional Ator Escoteiro

Este contexto permite visualizar que um usuário, identificado como escoteiro, pode acessar:

- a relação de dirigente em ordem alfabética;
- a relação de chefes por seção;
- a relação de escoteiros por seção;
- a relação de atividades por seção;
- as atividades realizadas por este escoteiro, bem como, a sua ficha 120.

Após a elaboração dos contextos navegacionais, inicia-se a confecção dos cartões de visão, que irão auxiliar na montagem do esquema navegacional. Os cartões de visão consistem em mapear todas as classes e contextos presentes em cada contexto navegacional, e, a partir deste mapeamento, localizar os nós e relacionamentos do esquema navegacional.

Abaixo, dois dos cartões de visão identificado a partir dos contextos navegacionais elaborados. A relação completa destes cartões de visão está no Anexo I deste trabalho.

Tabela 5: Exemplos de Cartões de Visão

<p>Ficha 120 por Escoteiro</p> <pre>from ge:GE, es:Escoteiro, ct:Conquista, pg:Progressão, ep:Especialidade, fc:Ficha120 ge.nome, ge.região, ge.numeral, ge.setor, ge.url, ge.e-mail, es.registro, es.nome, es.endereço, es.bairro, es.cidade, es.estado, es.CEP, es.telefone, es.celular, es.e-mail, es.data.nascimento, es.naturalidade, es.religião, es.pai, es.profissão.pai, es.telefone.pai, es.mãe, es.profissão.mãe, es.telefone.mãe, es.foto prog-nome: List of<pg.nome where fc tem pg> prog-data: List of<pg.data where fc tem pg> conq-nome: List of<ct.nome where fc tem ct> conq-data: List of<ct.data where fc tem ct> esp-nome: List of<ep.nome where fc tem ep> esp-data: List of<ep.data where fc tem ep> esp-nível: List of<ep.nível where fc tem ep> esp-área: List of<ep.área where fc tem ep> esp-tarefas: List of<ep.tarefas where fc tem ep> atividades: Idx Atividades por Escoteiro (es) imprimirFicha()</pre> <p>Escoteiro Leitura</p>
<p>Atividades por Seção Atividades por Escoteiro</p> <pre>observação, dias, mês, ano from at:Atividade, la:LocalAtividade, ta:TipoAtividade ramos: list of <ra:Ramo, ra.nome where ra participa at> local: la.nome where at acontece la tipo: ta.nome where at foi ta escoteiros: Idx Escoteiro por Seção (self)</pre> <p>Escoteiro Leitura</p>

6.4.2 Esquema Navegacional

Para cada grupo de cartões de visão, referentes a um contexto navegacional, é extraido um modelo navegacional.

Iniciando pelos cartões de visão do contexto navegacional do ator dirigente obtém-se oito nós navegacionais:

1 – Membro: Dirigente Alfabético – Chefe por Seção – Escoteiro por Seção;

2 – Membro: Dirigente Criação – Chefe Criação

3 – Membro: Dirigente Alteração

4 – Membro: Chefe Alterar Seção

5 – GE Alteração

6 – Seção Alfabético

7 – Seção Criação

8 – Seção Alteração

Da mesma forma que no modelo navegacional, neste momento, é aplicados os conceitos de orientação a objeto para definição das classes navegacionais (Nós). Gerando como resultado o diagrama navegacional do ator dirigente (Figura 39).

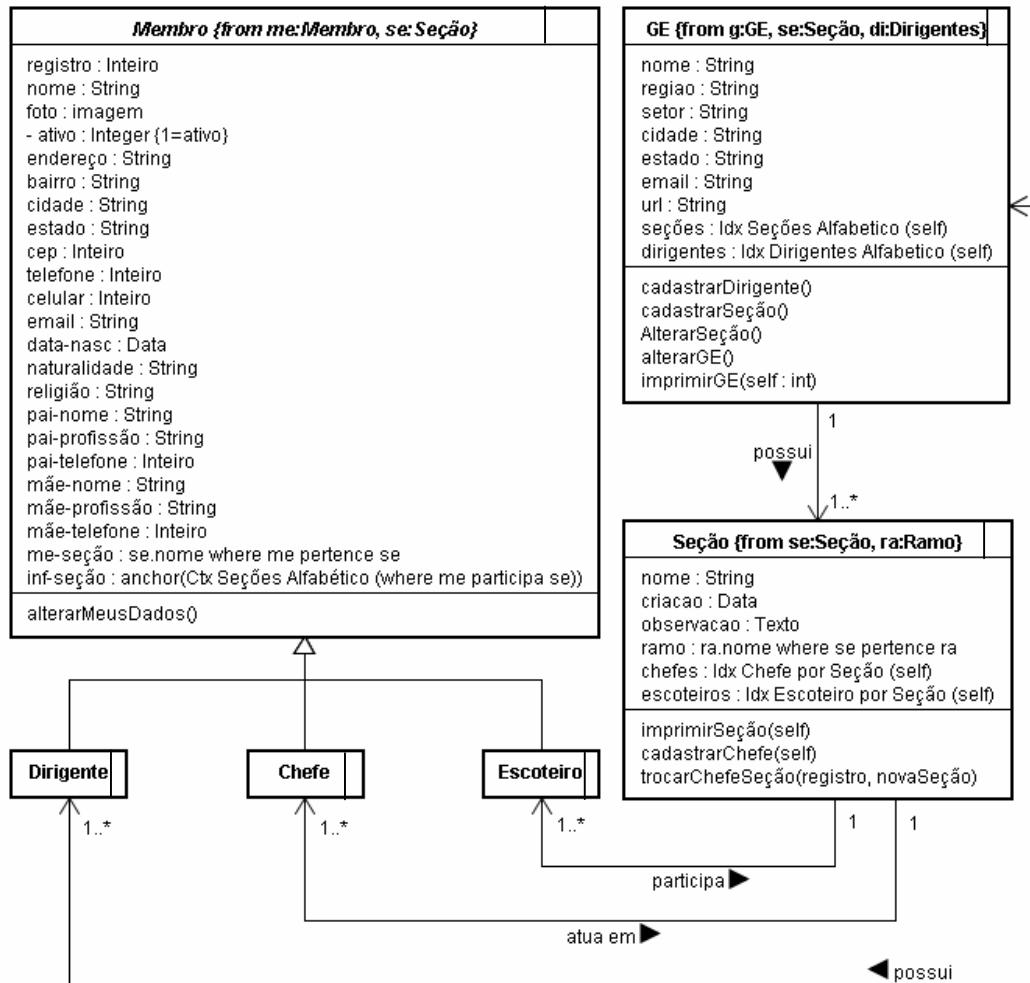


Figura 39: Diagrama Navegacional Ator Dirigente

Os mesmos procedimentos foram tomados para os cartões de visão dos chefes e escoteiros, resultados nos diagramas navegacionais de chefes e escoteiros respectivamente visualizados nas figuras 40 e 41.

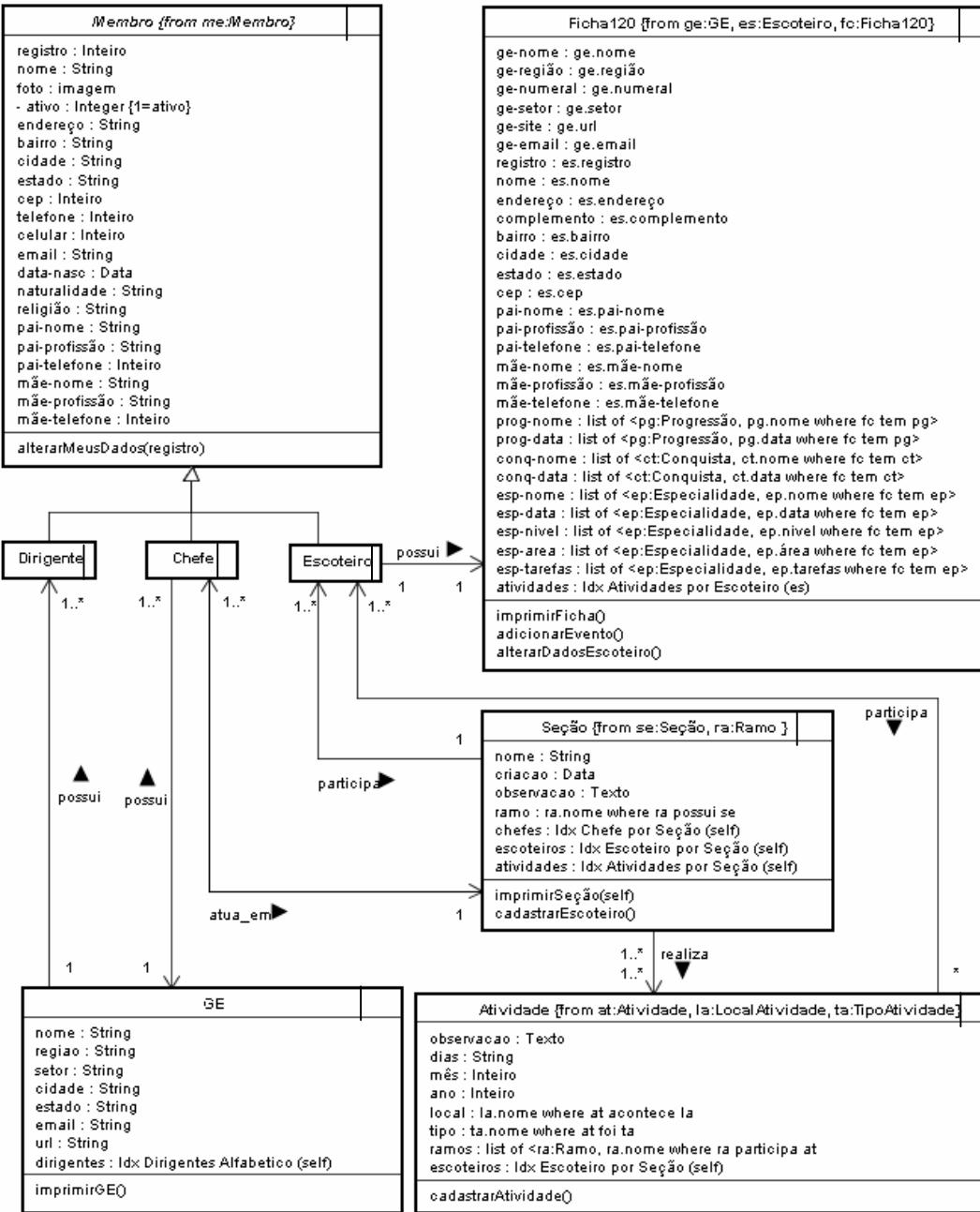


Figura 40: Diagrama Navegacional Ator Chefe

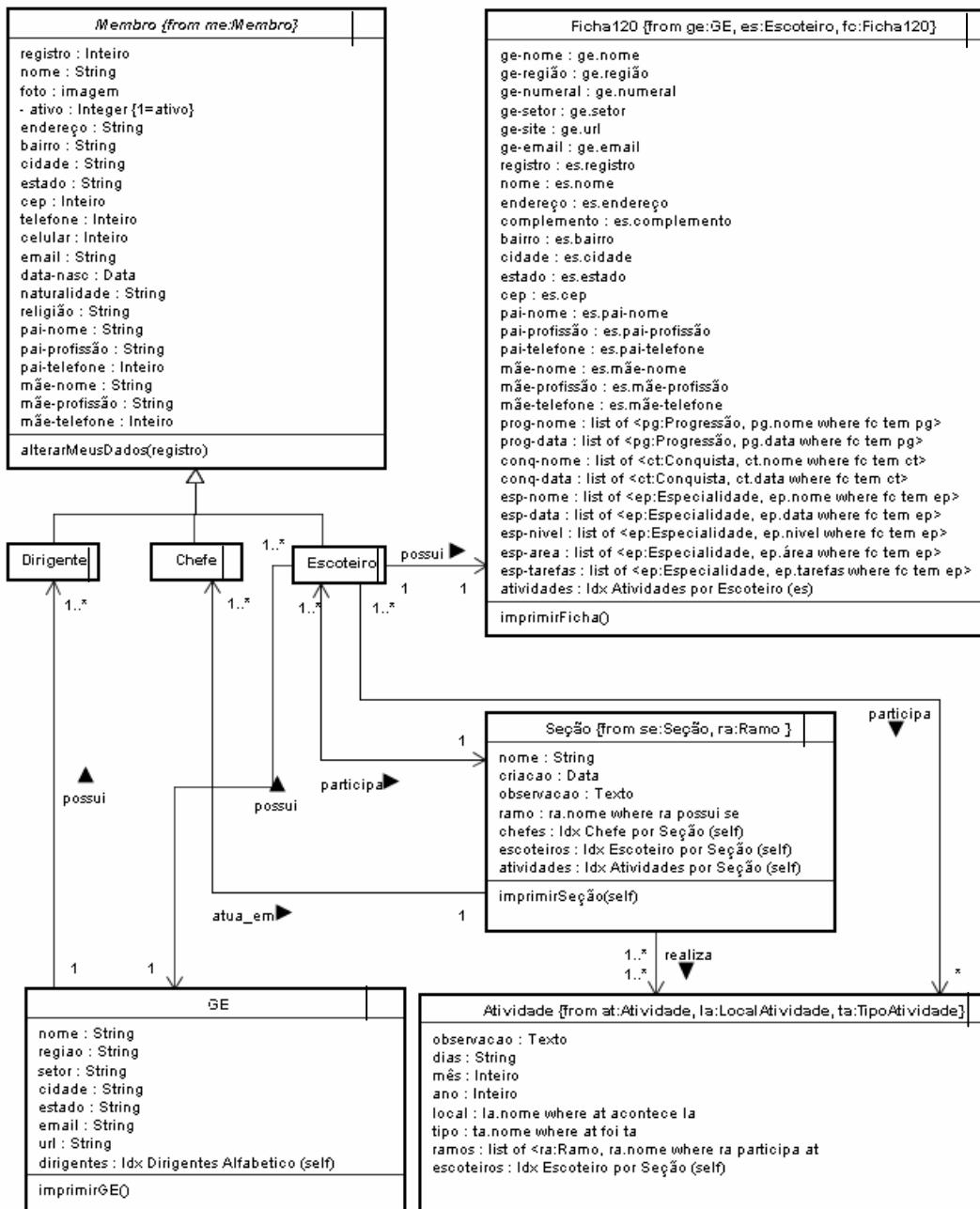


Figura 41: Diagrama Navegacional Ator Escoteiro

6.4.3 Cartões de Especificação

Como já visto no capítulo sobre OOHDM, os cartões de especificação são divididos em dois grupos (SHWABE & VILAIN, 2002):

- cartões de contexto: responsáveis por especificar as propriedades de um contexto; e
- cartões de estrutura de acesso: responsáveis por especificar as propriedades de uma estrutura de acesso.

Abaixo encontra-se um exemplo de cartão de contexto, bem como, de um cartão de estrutura de acesso da WebApp para o GE Ximbangue. Cada contexto, bem como, cada estrutura de acesso dos contextos de navegação modelados anteriormente estão disponíveis no Anexo I.

Tabela 6: Exemplo de Cartão de Contexto

Contexto: Chefe por Seção			
Parâmetros			
Elementos:	c:Chefe where c atua_em s:Seção		
Classe em Contexto:			
Ordenação	por c.nome, ascendente		
Navegação Interna:	por índice (Idx Chefe por Seção)		
Operações			
Usuário:	Dirigente	Permissão:	Leitura
	Chefe		Leitura
	Escoteiro		Leitura
Comentários:			

Tabela 7: Exemplo de Cartão de Estrutura de Acesso

Estrutura de Acesso: Seção: Chefes			
Nível			
Seção	Elementos: Dom (Seção)		
	Atributos s: Seção, s.nome, s.criação, s.observação, s.ramo	Destino Idx Chefes por Seção (s.nome)	
	Ordenação: por s.nome, ascendente		
	Parâmetros: s: Seção Elementos: c: Chefe where c.seção = s		
Chefes por Seção	Atributos c: Chefe, c.nome, c.registro, c.e-mail, c.telefone	Destino Ctx Chefes por Seção (c.nome)	
	Ordenação: por c.nome, ascendente		
	Usuário: Dirigente Chefe Escoteiro	Permissão: Leitura Leitura Leitura	
Comentários:			
Depende de:		Influencia:	

6.5 MODELAGEM DE INTERACE ABSTRATA

Para definir o projeto de interface são produzidos diagramas de configuração, compostos por ADVs e ADOs, e os ADVCharts (SHWABE & VILAIN, 2002).

Hennrichs (2005) define três fases para a realização da atividade de projeto de interface:

- Primeiro: identificação dos ADVs;
- Segundo: elaboração dos diagramas de configuração;
- Terceiro: especificação dos ADVCharts.

Os ADVs, segundo Hennrichs (2005), são responsáveis por exibir o modelo da interface abstrata, porém, estes objetos não conseguem interagir sozinho com o restante da aplicação, necessitando para isso, do auxílio dos ADOs (Abstract Data Objects – Objetos de dados abstratos) que irão comunicar a estrutura dos ADVs com seus devidos nós navegacionais, gerando no final, a exposição das informações contidas nos nós navegacionais e também permitindo o acesso a novos nós.

Abaixo, estão relacionados alguns ADVs identificados para a WebApp. A relação completa destes ADVs se encontra no Anexo I deste trabalho.



Figura 42: ADV Principal da WebApp.



Figura 43: ADV Conteúdo da WebApp.

Identificados os ADVs monta-se os diagramas de configuração conforme modelo abaixo. A relação completa se encontra no Anexo I.

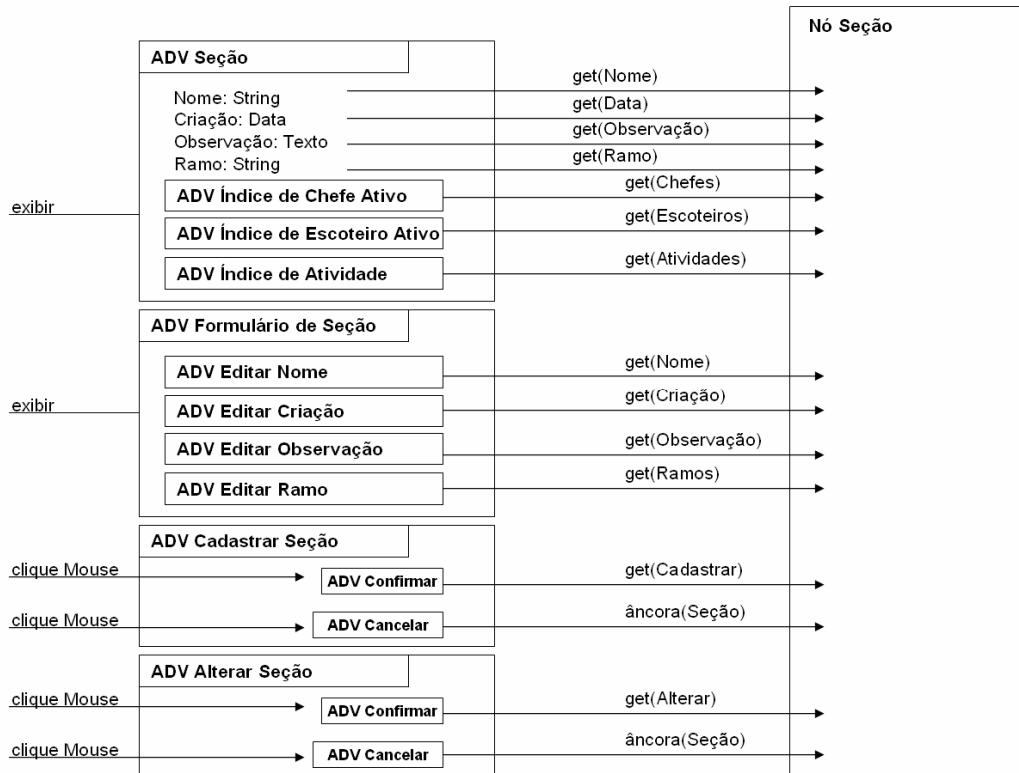


Figura 44: Diagrama de Configuração

No diagrama acima é possível a visualização dos ADOs responsáveis pela “instanciação” dos nós permitindo aos usuário a visualização dos conteúdos existentes nestes.

Os ADVCharts identificados na aplicação servirão de estrutura para a realização da autenticação do usuário por registro e senha ao sistema (Figura 45) e o controle de ativação e desativação de tarefas visualizadas no menu do usuário (Figura 46).

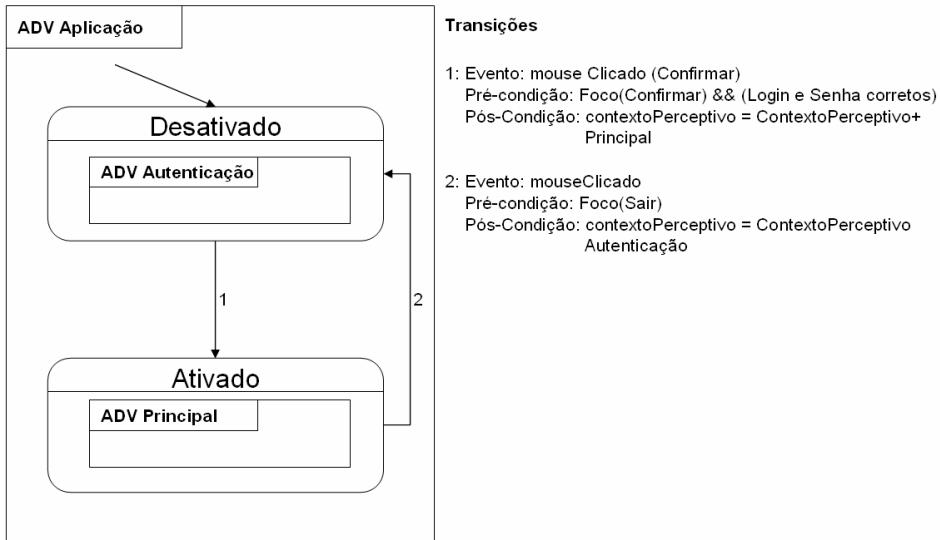


Figura 45: ADVCharts de Autenticação de usuário.

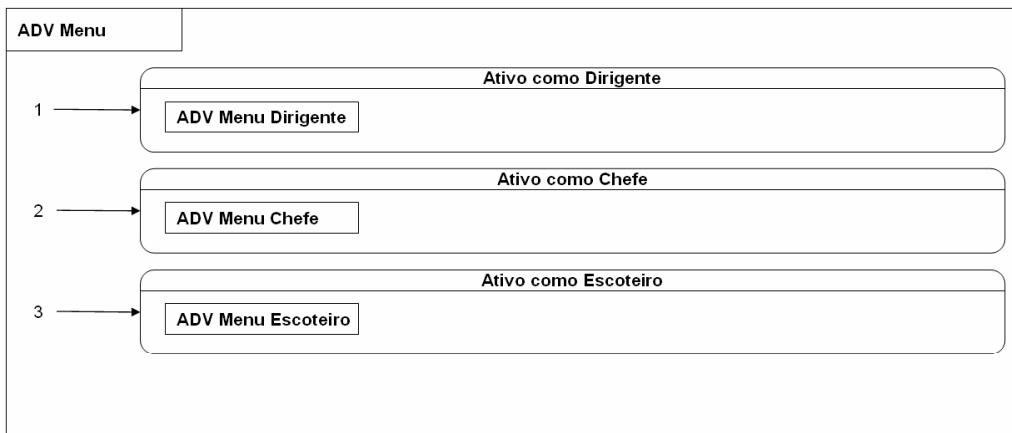


Figura 46: ADVCharts de Controle de menus.

6.6 IMPLEMENTAÇÃO

A fase de **implementação** (identificados em verde na figura 31) apresentada no OOHDM gera o agrupamento e resolução de todo material coletado pelas fases anteriores a esta. Permitindo com isso, o **desenvolvimento de protótipos** e aplicações cada vez mais aprimoradas, como também seus devidos **testes**.

O processo de implementação do OOHDM permite que esta aplicação não se restrinja a uma linguagem ou paradigma específico. Porém se torna mais tranquilo quando este processo é aplicado sobre um ambiente que suporte a orientação a objetos.

O desenvolvimento desta WebApp será realizado utilizando-se a linguagem PHP 5.0.2, e o Banco de Dados MySQL 4.x.. A escolha por esta linguagem e por este banco vem da fácil integração entre as duas tecnologias, por ambos serem *freeware*, ou seja, sem custos e pela familiaridade do autor deste trabalho com estes dois ambientes.

Detalhes de código fonte não estarão disponíveis nem serão, pois este, se restringe apenas aos processos documentacionais do OOHDM. Entretanto será apresentado adiante alguns screenshots da aplicação desenvolvida.

6.6.1 Mapeamento para modelo Relacional

A primeira atividade a ser relatada nesta fase do OOHDM é o mapeamento do modelo conceitual da aplicação para o modelo relacional de banco de dados.

As seguintes regras serão tomadas para se criar este modelo (SHWABE & VILAIN, 2002):

- Cada classe é mapeada para uma tabela relacional, obtendo-se a seguinte relação de tabelas: membro, dirigente, chefe, escoteiro, seção, ramo, GE, atividade, localAtividade, tipoAtividade, ficha120, evento, especialidade, progressão e conquista.
- Os relacionamentos de “*..*” (muitos para muitos) serão mapeados como tabelas relacionais: Escoteiro participa Atividade (atividadeEscoteiro), Ficha120 tem Atividade (fichaAtividade), Ficha120 tem Especialidade (fichaEspecialidade) e Ficha120 tem Progressão (fichaProgressao).

- Os atributos das classes são incorporados como atributos de suas tabelas de referência, bem como, os relacionamentos de “1..*” (um para muitos) acarretarão na incorporação, na tabela por parte de “*” (muitos), do identificador da tabela “1” (um).

A heurística pode ser aplicada sobre este processo, permitindo uma melhor viabilidade, facilidade e, principalmente, simplicidade de implementação.

Algumas alterações, propostas pela heurística, no modelo ER mapeado são:

- Transformar a especialização de Membro para Dirigente, Chefe e Escoteiro em uma tabela tipoMembro, definindo os três registros “0 – Dirigente”, “1 – Chefe” e “2 – Escoteiro”.
- Eliminar a tabela de Eventos, tornando suas especializações independentes.
- Migrar a data e nível (quando necessário), das tabelas de origem Especialidades, Progressões e Conquistas para suas respectivas tabelas relacionais à tabela de ficha120.
- Transformar o atributo áreas de Especialidade em uma tabela areaEspecialidade.
- Adicionar um campo senha na tabela Membro, a fim de identificá-lo dentro do sistema.

O modelo ER ficou estabelecido conforme a figura 47.

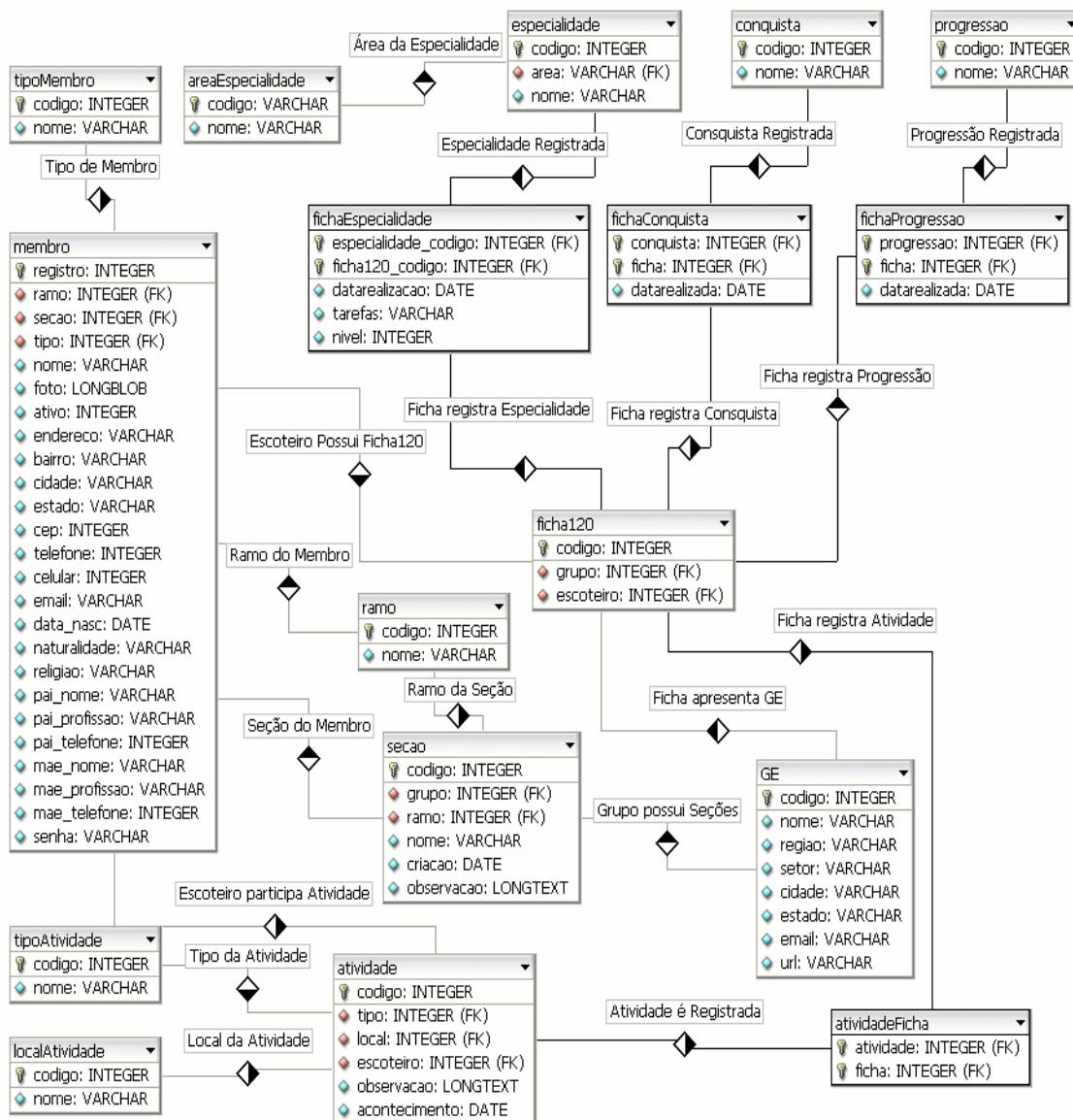


Figura 47: Diagrama de Modelagem-Relacionamento da WebApp.

6.6.2 Desenvolvimento e Testes

Realizado a modelagem do ER, parte-se para a geração das páginas e acessos ao banco de dados.

A implementação da aplicação fora desenvolvida sobre a arquitetura MVC vista no capítulo 2, dividindo a WebApp em três setores distintos:

- Modelo (Model): classes desenvolvidas em PHP implementam a lógica da aplicação, com base no diagrama de modelagem conceitual desenvolvido na modelagem conceitual. Nesta camada também ocorrem as solicitações ao banco de dados.
- Visualização (View): códigos editados em HTML geram os templates da aplicação, comumente conhecidos como gabaritos. Estes templates são montados de acordo com as especificações realizadas através dos ADVs da modelagem de interface abstrata.
- Controle (Control): também desenvolvido em PHP, assume o papel, tanto dos controles de requisição dos ADOs da modelagem de interface, como as permissões de contexto do modelo navegacional.

A implementação utilizando esta arquitetura permite, de forma clara, a aplicação de recursividade para novas aplicações.

Abaixo, segue alguns screenshots da aplicação desenvolvida:

The screenshot shows the homepage of the GE Ximbangue 21 - Chapecó SC website. At the top right is the logo 'Uma aventura Saudável'. The main content area features a banner with several young people outdoors. Below the banner are navigation links: 'GE', 'Seção', 'Atividades', 'Perfil', and 'Sair'. A sidebar on the left contains sections for 'Ações' (with 'Editar Grupo' and 'Criar Seção' buttons), 'Calendário' (showing a June 2005 calendar), and 'Meu Perfil' (with a placeholder 'Dirigente: Fulano de Tal'). The central content area is titled 'Grupo Escoteiro' and displays information for 'Grupo Escoteiro Ximbangue - 21 - Chapecó SC'. It includes fields for 'Endereço: Parque Verdes - Chapecó - SC', 'E-mail: ximbangue@ximbangue.com.br', and 'url: www.ximbangue.com.br'. Below this is a table titled 'Dirigentes' listing four members with their contact details. A section titled 'Seções' lists various scouting troops. At the bottom is a copyright notice: 'Copyright 2005 - Grupo Escoteiro Ximbangue Região 21 - Chapecó-SC'.

Figura 48: Screenshot Visualizar Grupo Escoteiro.

This screenshot shows the 'Editando Informações Institucionais do GE Ximbangue' page. The main content area contains a form with fields for 'Nome: Ximbangue', 'Endereço: Parque Municipal, Venda Bairro Efapi, CEP 89880-000', 'Região: 21', 'Setor: 16', 'Cidade: Chapecó', 'Estado: SC', 'E-mail: ximbangue@ximbangue.com.br', and 'URL: www.ximbangue.com.br'. Below the form are 'Confirmar' and 'Cancelar' buttons. The left sidebar and footer are identical to Figure 48.

Figura 49: Screenshot Editar Grupo Escoteiro.

GE Ximbangue 21 - Chapecó SC

Seção Ramo Escoteiro

Tropa Escoteira Fénix

Criado: 12/05/1983

Observações:

Metas para este ano:

- Destaque de melhor tropa de Santa Catarina
- Reativar a patrulha Águila
- Campanha das rosas

Chefs

Nome do Dirigente	Registro	Telefone	E-mail
Cristiano Popov Zambiasi	256462-4	323-4993	cristiano@ximbangue.com.br
Lara Popov Zambiasi	369154-1	323-4993	lara@ximbangue.com.br
Claudia	252354-6	323-1244	claudia@ximbangue.com.br
Saulo Bazzi Oberdorfer	487444-1	323-3032	saulo@ximbangue.com.br

Escoteiros

Nome do Dirigente	Registro	Telefone	E-mail
Amanda	256462-4	323-4993	amanda@ximbangue.com.br
Felipe	369154-1	323-4993	felipe@ximbangue.com.br
André	252354-6	323-1244	andre@ximbangue.com.br
Alexandre	487444-1	323-3032	alexandre@ximbangue.com.br
Giovana	554874-6	323-8456	giovana@ximbangue.com.br
Marta	777845-9	322-8414	marta@ximbangue.com.br

Copyright 2005 - Grupo Escoteiro Ximbangue Região 21 - Chapecó-SC
Todos direitos autorais reservados.

Figura 50: Screenshot Visualizar Seção.

GE Ximbangue 21 - Chapecó SC

Cadastrar Nova Atividade

Cadastrar Atividade para Tropa Escoteira Fénix

Últimas atividades realizadas:

02 do 06/2005	Visita	Museu Municipal Chapecó	Lobinho
02 do 06/2005	Jornada	Cede Campestre do CRC	Escoteiro
02 e 03 do 06/2005	Acampamento	Trilha do Pitoco	Sênior/Pioneiro

Cadastrar nova atividade:

Local:	Sede Ximbangue	Novo Local:	<input type="text"/>
Tipo:	Acampamento	Novo Tipo:	<input type="text"/>
Dias:	<input type="text"/>	Mês/Ano:	<input type="text"/> / <input type="text"/>

Observações:

Escoteiros Participantes

Nome do Escoteiro

Amanda

Felipe

André

Alexandre

Giovana

Marta

Confimar **Cancelar**

Copyright 2005 - Grupo Escoteiro Ximbangue Região 21 - Chapecó-SC
Todos direitos autorais reservados.

Figura 51: Screenshot Cadastrar Atividade.

6.7 AVALIAÇÃO E MANUTENÇÃO

A **avaliação e manutenção** (identificados em laranja na figura 31), fase não declarada diretamente na metodologia OOHDM, funciona como uma âncora, um re-contato entre a WebApp criada e os atores e tarefas identificados na primeira fase (Planejamento). É a forma que o OOHDM tem de reencontrar a solução proposta e implementada ao seu problema inicial. Aspecto muito necessário na engenharia web segundo Pressman (2002).

Avaliar com o cliente os resultados obtidos através da engenharia é fator fundamental no desenvolvimento de uma aplicação. Permitindo tanto para o cliente, como para o desenvolvedor, a revisão do projeto, de forma a encontrar estados crítico de atividades, possíveis gerações de erros e, principalmente, as novas necessidades que a aplicação deve auxiliar, tornando esta fase, não o fim de uma engenharia, mas o início de um novo projeto, melhor e maior do que o desenvolvido até agora (PRESSMAN, 2002).

O protótipo da WebApp foi publicado no domínio web www.ximbangue.com.br para sua avaliação. Sendo este, testado por chefes, dirigentes e escoteiros do GE Ximbangue.

Por sua engenharia bem definida, não houve divergências ou solicitação de outras funcionalidades em relação aos requisitos que a WebApp se propusera. Sendo que para esta proposta, o protótipo se mantivera fiel as informações geradas pelo método OOHDM e estas informações estavam concisas com as reais necessidades do GE Ximbangue.

7 CONSIDERAÇÕES FINAIS

Desenvolver uma WebApp através da engenharia web utilizando o método OOHDM é uma tarefa desafiadora, possibilitando perceber fatos, que sem a aplicação das atividades propostas por este método, passariam despercebidos em um projeto “ad hoc” (sem modelos de processo). Porém, como já visto, não há uma melhor metodologia para ser aplicada a uma aplicação qualquer, mas sim, há metodologias que melhor se aplica a uma determinada abordagem de uma aplicação.

Tratando-se de aplicações web, mais diretamente WebApps, as escolha do OOHDM para abordar a estratégia de engenharia web proposta por Pressman e Ginige permite claramente vislumbrar que esta metodologia possui uma robustez muito elevada em relação as informações a serem tratadas e armazenadas na elaboração de projetos hipermídia.

7.1 RESUMO DAS CONTRIBUIÇÕES

A aplicação de uma engenharia web para o caso real de desenvolvimento de uma WebApp para um Grupo escoteiro através do método OOHDM permitiu obter os seguintes resultados:

- o OOHDM implementou todos passos, visíveis nesta pesquisa, para o desenvolvimento de uma WebApp com a aplicação de conceitos de engenharia web em uma estrutura de desenvolvimento incremental e prototipada;
- o OOHDM ainda pode oferecer riscos, quando utilizado para realização de projetos comerciais, pois necessita de muitas interações entre os projetistas e os usuários que irão utilizar a aplicação, podendo gerar, quando não completamente dominado por parte dos projetistas, em um provável atraso de projeto;
- a engenharia web difere da engenharia tradicional em poucos aspectos, o que encoraja a desenvolver aplicações web que atendam cada vez mais aos critérios de engenharia, concordando que este tipo de desenvolvimento não é uma obra de arte ou um artigo de publicidade, mas sim um software que necessita de atenção especiais e que podem ser facilmente elaborados utilizando-se as ferramentas corretas;

- o OOHDM não só oferece uma interação amigável a engenharia web, como também, aos ambientes e arquiteturas de desenvolvimento existentes atualmente e aos colaboradores que necessitam das documentações geradas por ele para identificar o foco e o escopo do projeto e da aplicação;
- não somente o OOHDM, mas também o Escotismo e a estrutura organizacional de um grupo escoteiro possuem uma arquitetura de procedimentos e tarefas bem definidos, elaborados e de fácil entendimento;

7.2 TRABALHOS FUTUROS

O OOHDM apesar de sua robustez está ainda no início de sua carreira no mundo da tecnologia Web-Hipermídia. Sendo assim, ainda há muita coisa a ser feita, como:

- Uma ferramenta CASE que auxilie nas modelagens e na geração das documentações;
- O desenvolvimento de técnicas para o gerenciamento das aplicações desenvolvidos sobre perfis de usuários, permitindo uma maior interação entre o que o modelo oferece e o conteúdo que esta sendo trabalhado nele;
- Um estudo mais aprofundado nas fases de navegação e interface abstrata, talvez adaptando o material proposto pelo OOHDM à engenharia de usabilidade, gerando mais significado ao uso desse material.

Quanto a Engenharia Web, muito já vem se estudando e desenvolvendo sobre essa área, a cada dia vem surgindo novas técnicas, para facilitar o processo, e novas justificativas para incentivar o uso desta atividade. Há um padrão muito alto nas bibliografias sobre o modelo evidenciado por Pressman e Ginige sobre esse tópico da engenharia. Um estudo sobre outras formas de se aplicar engenharia sobre aplicações web sempre serão bem vindas ao meio acadêmico, nem que seja para fazer um comparativo com as já existentes, demonstrando o porquê desta filosofia de engenharia web ser tão seguida.

No escotismo, muito há a ser feito:

- Evoluir a aplicação proposta neste trabalho para um dicionário de atividades escoteiras;
- Desenvolver aplicações web que auxiliem na comunicação entre diferentes seções ou diferentes grupos;
- Estender esta aplicação para englobar um número maior de grupos escoteiros, permitindo talvez, a aplicação de ferramentas mineradoras para extrair conceitos educacionais de atividades mais regionalizados.

O escotismo tem suas portas e janelas abertas a todos que queiram melhorar, de forma voluntária, o movimento escoteiro e o mundo de forma geral, então, neste caso, as sugestões servem apenas de base para vislumbrar todos os benefícios que a tecnologia pode trazer ao movimento escoteiro.

8 REFERÊNCIAS

- ÁLVARES, Patrícia M., **WebPraxis – Um processo personalizado para projetos de desenvolvimento para a Web**, IV Semana de Pós-Graduação, UFMG, 2000.
- ALVES, Willian P., **Crie, Anime e Publique seu Site Utilizando Fireworks Mx 2004, Flash Mx 2004, Dreamweaver Mx 2004**, 1ª Edição, São Paulo, Érica, 2004.
- AURÉLIO, Dicionário Eletrônico**, versão 3.0, Lexicon Informática Ltda., nov. 1999.
- Baden Powell, **Guia do Chefe Escoteiro**, Editora Escoteira, 4 edição, 1982
- BENYON, D. R., GREE, T. R. G. and Bentall, D., **Conceptual Modelling for Human Computer Interaction, using ERMIA** Springer, London, 1996
- BOLCHINI, Davide, **Web Design Patterns: Improving Quality and Performance in Web Application Design**. Tese (Doutorado em Ciência da Comunicação) – Facolt di scienze della comunicazione Lugano, Universidade de Svizzera Italiana, Lugano, 2000.
- BREVE, Fabrício Aparecido, **Engenharia para a Web**. Disponível em: <http://www.fabricio.aidi.com.br/trabalhos/engenharia_web.pdf> Acessado em: set. 2004.
- CARDOSO, André. **O PHP nasce de novo**. Revista Info Exame, São Paulo, v. 222, p.99, Set. 2004.
- CASTAGNETTO, Jesus, et al., **Professional PHP: Programando**, 1ª Edição, São Paulo, Makron Books, 2001.
- CONALLEN, Jim. **Desenvolvendo Aplicações Web com UML**, Rio de Janeiro, Campus, 2003.
- CONTENT MANAGEMENT, **What is Content?**. Disponível em: <<http://www.contentmanager.eu.com/history.htm>> Acessado em: nov. 2004.
- COSTAGLIOLA, Gennaro, FERRUCI, Filomena, FRANCESE, Rita. **Web Engineering: Models and Methodologies for the Design of Hypermedia Applications**, Departamento de Matemática e Informática da Universidade de Salerno, Baronise (SA). Itália, 2002.

EPNER, M., Poor Project Management Number-One Problem of Outsourced E-Projects, Research Briefs, Cutter Consortium, 2000.

Estatuto da União dos Escoteiros do Brasil, 15 e 16 de novembro de 2003. Disponível em: <www.ueb-df.org.br/Publicacoes/ESTATUTO_2003_DA_UEB.doc> Acesso em: 20 nov. 2004.

FAGUNDES, Flávio Z., Desenvolvimento de Aplicações Web, 2004. Disponível em: <<http://www.vivaolinux.com.br/artigos/impressora.php?codigo=815>> Acesso em: mar 2005.

FARINON, Márcio et al, De Lobinho a Pioneiro: a criança e o jovem com quem trabalhamos, Brasília, abcBSB, 1995.

GINIGE, Athula, Web Engineering: Managing the Complexity of Web Systems Development. University of Western Sydney, Itália, 2002.

HENNRICH, Jean Carlos, Estudo da Metodologia Orientada a Objetos OOHDIM, para Modelagem e Desenvolvimento de Websites. Monografia (Pós-Graduação em Ciência da Computação) – Centro de Tecnologia e Ciência, Universidade Federal de Santa Catarina, Florianópolis, 2005.

HENNRICH, Jean Carlos, Modelagem e Desenvolvimento de um Aplicativo Hipermídia sobre o RMS Titanic usando a Metodologia OOHDIM e Software de Autoria. Monografia (Bacharelado em Ciência da Computação) – Centro de Tecnologia da Universidade do Oeste de Santa Catarina, Chapecó, 2000.

HORN, Luiz C., Curso de Nível Avançado Ramo Escotista, UEB-SC. Florianópolis, set. 2004.

INFOIMAGEM, Gestão de quê?... Documentos? Conteúdos? Conhecimento?, 2001. Disponível em: <<http://www.doc-imagem.com/infoimagem/info33/33art1.htm>> Acesso em: mar 2005.

ISAKOWITZ, Tomás, EDWARD, A., BALASUBRAMANIAN, P., RMM: A Methodology for Structured Hypermedia Design. Leonard N. Stern School of Business, EUA, 1995. Disponível em: <<http://citeseer.ist.psu.edu/cache/papers/cs/678/http:zSzSzwww-is.informatik.uni-oldenburg.dezSz~dibozSzteachingzSzmmzSzlinkszSz..zSzpageszSzrmd.pdf/isakowitz95rmm.pdf>> Acessado em: nov 2004.

KAPPEL, Gerti et al, **WEB ENGINEERING – Old wine in new bottles?**, 2004. Disponível em: < <http://www.big.tuwien.ac.at/research/publications/2004/0404.pdf>> Acesso em: nov 2004.

LOCATELLI, Márcio Henrique. **Engenharia de Software para o Desenvolvimento de WebApps e as Metodologias OOHDM e WebML**. Monografia (Pós-Graduação em Ciência da Computação) – Centro de Tecnologia e Ciência, Universidade Federal de Santa Catarina, Florianópolis, 2003.

MEDEIROS, Adriana P., SCHWABE, Daniel, **Especificação Declarativa de Aplicações Web em OOHDM**, PUC-Rio, 2001.

MOLINARI, Leonardo, **Gestão de Projetos: Técnicas e Práticas com Ênfase em Web**, 1ª Edição, São Paulo, Érica, 2004.

MORATELLI, Alexandre S., VALDAMERI, Alexander R., **Sistema de Gerenciamento de Conteúdo para Ambiente Web**. FURB, Brasil, 2002.

NAGY, Lazlo. **250 Milhões de Escoteiros**. The World Scout Association, Geneva, Switzerland, 1985.

NIELSEN, Jakob, **Projetando Websites**, 1ª Edição, Rio de Janeiro, Campus, 2000.

OLSINA, Luiz, **Web-site Quality Evaluation Method: a Case Study on Museums**, UNLPan, 1999.

P.O.R., Princípios, Organizações e Regras, resolução de nível nacional da UEB, 2001.

PAIVA, Débora, **Desenvolvimento de Software para Web: Conceitos Fundamentais e um Estudo de Caso**, Palestra (Dia da Computação do CURSO DE CIÊNCIA DA COMPUTAÇÃO UNOCHAPECÓ). Chapecó, out. 2003.

PRESSMAN, Roger S., **Engenharia de Software**, 5ª Edição, Rio de Janeiro, McGraw-Hill, 2002.

PRESSMAN, Roger S., **Engenharia de Software**, 2ª Edição, Rio de Janeiro, McGraw-Hill, 1995.

ROQUE, Ruth F., **ESTUDO COMPARATIVO DE METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO UTILIZANDO A TÉCNICA DELPHI.** Monografia (Pós-Graduação em Ciência da Computação) – Centro de Tecnologia e Ciência, Universidade Federal de Santa Catarina, Florianópolis, 1998. Disponível em: < <http://www.eps.ufsc.br/disserta98/ruth/>>.

ROSSI, Gustavo, SCHWABE, Daniel, **An Object Oriented Approach to Web-Based Application Design.** PUC-RIO, Brasil, 1998.

SCHWABE, Daniel, PONTES, Rita, MOURA, Isabela, **OOHDM-Web: An Environment for Implementation of Hypermedia Applications in the WWW.** PUC-Rio, 1999.

SCHWABE, Daniel, **Resumo do OOHDM.** Disponível em: < <http://www.oohdm.inf.puc-rio.br:8668/space/resumo+do+OOHDM>> Acessado em: out. 2004.

SCHWABE, Daniel, VILAIN, Patrícia, **Notação do Método OOHDM** Versão 2.0, PUC-Rio, 2002.

SCHWENDIMAN, Blake., **PHP4: Guia do Programador,** 1^a Edição, Rio de Janeiro, Ciéncia Moderna Ltda., 2001.

SILVA, Lizandro K. da, et al, **Um Assistente Digital para Responder Automaticamente Perguntas de Usuários Humanos em Portais Corporativos,** 2003, Diponível em: < <http://www.adsdigital.com.br/arquivos/iskm2003ad.pdf>>. Acessado em: mar. 2005.

UML, UML Resource Center. Disponível em: < <http://www.rational.com/uml>> Acessado em: nov. 2004.

W3C, World Wide Web Consortium. Disponível em: <www.w3c.org>. Acessado em abr. 2005.

WebEstilo, Usabilidad, Programation y Mucho más. Disponível em: <<http://www.webestilo.com>> Acessado em: jun. 2005.

WIZARD STRUTS, Wizard STRUTS, Disponível em: < <http://www.qualiti.com.br/arquivos/coder/manualWizardStruts.PDF>> Acessado em: abr. 2005.

9 ANEXOS

Relação de documentos gerados pelo método OOHD para o desenvolvimento de uma WebApp de controle de fichas 120 para o GE Ximbangue.

9.1 RELAÇÃO DE CENÁRIOS

Cenário: C1.1 (Dirigente) – Alterar informações institucionais do GE.

Contexto: Sou dirigente de um grupo escoteiro e necessito que todas as Fichas 120 dos escoteiros deste grupo sejam visualizadas e impressas com as informações cadastrais atuais do grupo escoteiro.

Objetivo: Atualizar os dados cadastrais referentes ao grupo escoteiro.

Ações: Eu acesso a área de gerenciamento de informações cadastrais do GE. Insiro as informações institucionais, solicitadas pelo sistema, pertinentes ao grupo escoteiro como seu nome, sua região, seu numeral e setor, bem como, cidade e estado, e-mail e o endereço do site do GE e escolho a opção de atualizar os dados que acabei de inserir.

Cenário: C1.2 (Dirigente) – Cadastrar uma nova seção no GE.

Contexto: Sou dirigente de um grupo escoteiro e acabamos de inaugurar uma nova alcatéia devido a grande procura de crianças pelo movimento escoteiro e a disponibilidade de chefes no GE. Precisamos então criar essa nova seção para que possam ser transferidos os devidos chefes para esta, bem como, os novos membros juvenis.

Objetivo: Inserir uma nova seção no GE.

Ações: Eu acesso a área de gerenciamento de seções e escolho a opção de criação de nova seção. Informo o nome e o ramo que esta nova seção possuirá, bem como, a data de criação e algumas observações, caso necessário. Confirmo os dados que acabei de inserir e o sistema realiza o cadastro desta nova seção no GE.

Cenário: C1.3 (Dirigente) – Cadastrar um novo chefe em uma seção.

Contexto: Acabamos de inaugurar uma nova alcatéia devido a grande procura de crianças pelo movimento escoteiro e a disponibilidade de chefes no GE. Porém ainda necessitamos de chefes e alguns pais destes novos jovens resolveram aceitar o desafio e se disponibilizaram a ser chefes para auxiliar o movimento.

Objetivo: Cadastrar um novo chefe em uma seção.

Ações: Eu acesso a área de gerenciamento de seções e escolho a opção de criação de um novo chefe. Informo o nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento do chefe, bem como, sua naturalidade e religião. Procuro no meu computador e seleciono uma foto deste chefe, registrada em sua inscrição. Seleciono a seção em que o novo chefe vai participar. Confirmo os dados que acabei de inserir e o sistema realiza o cadastro do chefe.

Cenário: C1.4 (Dirigente) – Ver escoteiros por seção.

Contexto: Sou dirigente de um grupo escoteiro e preciso de uma relação de todos os escoteiros ativos no grupo, por seção, para que eu possa refazer os registros de final de ano com a UEB.

Objetivo: Visualizar relação de escoteiros ativos existentes no GE.

Ações: Eu acesso a área de escoteiros por seção e o sistema me relaciona a relação de seções presentes no grupo. Eu seleciono uma seção e o sistema me retorna a relação dos escoteiros presentes atualmente nesta seção com seu registro, nome, seção e ramo, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um escoteiro e ver sua ficha 120 (ver ficha 120 de um escoteiro). Eu posso ainda retornar a lista e consultar os dados de outro membro.

<p>Cenário: C1.5 (Dirigente) – Ver a ficha 120 de um escoteiro.</p>
<p>Contexto: Sou dirigente do GE e desejo obter mais informações sobre o escoteiro selecionado.</p>
<p>Objetivo: Visualizar Ficha 120 de um escoteiro.</p>
<p>Ações: Eu seleciono o escoteiro pelo seu nome e o sistema apresenta os dados deste escoteiro no formato de ficha 120 com seu nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento, sua naturalidade e religião. Exibe ainda os nomes, profissões e telefones comerciais dos pais. A foto deste membro, a seção e o ramo as quais o membro pertence. A ficha ainda apresenta: o nome, a região, o numeral, o setor, o site e o e-mail do grupo escoteiro cadastrado no sistema; as datas de cada conquista atingida pelo escoteiro; as datas de cada progressão alcançada pelo escoteiro; a data e o nível de cada especialidade conquistada pelo jovem, bem como os índices da seqüência de tarefas que ele realizou; a data, o tipo, o local, os ramos e as observações de cada atividade realizada pelo escoteiro em sua vida escoteira. E ainda posso imprimir essas ficha.</p>

<p>Cenário: C1.6 (Dirigente) – Ver chefes por seção.</p>
<p>Contexto: Sou dirigente de um grupo escoteiro e preciso de uma relação de todos os chefes ativos no grupo, por seção, para que eu possa refazer os registros de final de ano com a UEB.</p>
<p>Objetivo: Visualizar relação de chefes ativos existentes no GE.</p>
<p>Ações: Eu acesso a área de chefes por seção e o sistema me relaciona a relação de seções presentes no grupo. Eu seleciono uma seção e o sistema me retorna a relação dos chefes presentes atualmente nesta seção com seu registro, nome, seção e ramo, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um chefe para ver mais informações (ver detalhes de um membro). Eu posso ainda retornar a lista e consultar os dados de outro membro.</p>

Cenário:	C1.7 (Dirigente) – Reativar um membro no GE.
Contexto:	Há um ano atrás um lobinho teve que ser desativado do GE pois estava de mudança com sua família para outra cidade. Porém, nesta última semana, este jovem voltou para Chapecó com seus pais e quer retornar ao GE Ximbangue. Então preciso reativá-lo para que ele possa voltar a receber os registros de atividades, conquistas e especialidades que irá fazer.
Objetivo:	Reativar um membro no GE.
Ações:	Eu acesso a área de gerenciamento de membros do GE e seleciono a visualização de membros desativados. O sistema me retorna a relação dos membros desativados no grupo com seu registro, nome, seção e ramo. Eu seleciono o membro que desejo reativar e o sistema me retorna a relação dos membros ativos com a presença do membro que acabei de selecionar. Caso a seção do membro não exista mais, o sistema solicita que eu escolha a nova seção (existente atualmente) para que este faça parte.

Cenário:	C1.8 (Dirigente) – Trocar um chefe de seção
Contexto:	Acabamos de inaugurar uma nova alcatéia devido a grande procura de crianças pelo movimento escoteiro e a disponibilidade de chefes no GE. Para iniciar-mos as atividades desta alcatéia necessitamos, além da entrada de novos chefes, da participação de chefes com maior experiência, já existentes na outra alcatéia.
Objetivo:	Alterar a seção de um chefe.
Ações:	Eu acesso a área de gerenciamento de seções e escolho a opção de transferência de chefe entre seções. O sistema me apresenta a relação de chefes ativos presentes no grupo com seu nome, registro na UEB, seção e ramo. Eu seleciono o chefe que desejo transferir e escolho a nova seção em que este chefe vai participar.

Cenário: C1.9 (Dirigente) – Cadastrar um novo dirigente no GE.

Contexto: Estamos finalizando o ano, e com ele, o mandato de dois anos desta diretoria. É nosso dever dar suporte e assistência a nova diretoria que irá assumir suas atividades neste próximo ano. Para que estes novos voluntários dirigentes possam ter acesso às informações do sistema é necessário que a diretoria atual os cadastre no sistema.

Objetivo: Cadastrar um novo dirigente em uma seção.

Ações: Eu acesso a área de gerenciamento de dirigentes e escolho a opção de criação de um novo dirigente. Informo o nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento do dirigente, bem como, sua naturalidade e religião. Procuro no meu computador e seleciono uma foto deste dirigente, registrada em sua inscrição. Confirmo os dados que acabei de inserir e o sistema realiza o cadastro do dirigente.

Cenário: C1.10 (Dirigente) – Ver dirigentes presentes no GE.

Contexto: Sou dirigente de um grupo escoteiro e preciso da relação de todos os dirigentes ativos no grupo, para agendar uma reunião.

Objetivo: Visualizar relação de dirigentes ativos existentes no GE.

Ações: Eu acesso a área de gerenciamento de dirigentes e o sistema me retorna a relação dos chefes presentes atualmente com seu registro, nome, telefone e e-mail, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um dirigente para ver mais informações (ver detalhes de um membro). Eu posso ainda retornar a lista e consultar os dados de outro membro.

Cenário: C1.11 (Dirigente) – Ver detalhes de um membro.

Contexto: Estou visualizando a relação de membros adultos presentes no GE (Chefes ou Dirigentes) e desejo ver mais informações sobre este membro.

Objetivo: Visualizar informações de um membro.

Ações: Eu seleciono o membro pelo nome e o sistema me retorna o nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento do membro, bem como, sua naturalidade e religião. Apresenta ainda sua foto, seu ramo e sua seção, caso pertença a uma.

Cenário: C1.12 (Dirigente) – Alterar alguns campos dos meus dados cadastrais.

Contexto: Estou mudando de endereço e também de provedor de internet. Para que o sistema do GE continue atualizado, preciso alterar as minhas informações cadastrais.

Objetivo: Altera dados cadastrais de um membro.

Ações: Eu acesso a área de atualização de perfil de usuário. O sistema me retorna a relação de campos que eu posso alterar, em relação aos meus dados cadastrais como: endereço, bairro, cidade e CEP, telefone, celular, e-mail e religião. Eu escolho o campo que alterar e insiro o valor atualizado, após isso, confirmo minha alteração.

<p>Cenário: C2.1 (Chefe) – Ver todos os escoteiros de sua seção.</p>
<p>Contexto: Sou chefe da seção escoteira, e estamos planejando uma atividade fora da Sede. Para isto preciso da relação de escoteiros pra providenciar as autorizações e confirmar as chamadas.</p>
<p>Objetivo: Visualizar relação dos escoteiros ativos existentes em uma seção.</p>
<p>Ações: Eu acesso a área de gerenciamento de escoteiros. O sistema me retorna a relação dos membros presentes atualmente na minha seção com seu registro, nome, telefone e e-mail, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um membro e ver sua ficha 120 (Ver e imprimir a ficha 120 de um escoteiro de sua seção). Eu posso ainda retornar a lista e consultar os dados de outro membro.</p>

<p>Cenário: C2.2 (Chefe) – Ver os chefes pertencentes a sua seção.</p>
<p>Contexto: Sou chefe da seção escoteira e preciso entrar em contato com os outros chefes de minha seção para agendar uma reunião e discutir detalhes sobre o acampamento que ocorrerá no próximo fim de semana.</p>
<p>Objetivo: Visualizar relação dos chefes ativos existentes em uma seção.</p>
<p>Ações: Eu acesso a área de gerenciamento de chefes. O sistema me retorna a relação dos chefes presentes atualmente na minha seção com seu registro, nome, telefone e e-mail, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um membro e ver mais informações sobre o mesmo (Ver detalhes de um membro). Eu posso ainda retornar a lista e consultar os dados de outro membro.</p>

Cenário: C2.3 (Chefe) – Cadastrar um novo escoteiro em sua seção.

Contexto: Sou chefe de um grupo escoteiro e é de minha responsabilidade cadastrar o novo membro juvenil que acabou de ingressar na minha seção.

Objetivo: Cadastrar um novo escoteiro em uma seção.

Ações: Eu acesso a área de gerenciamento de escoteiros e escolho a opção de criação de um novo escoteiro. Informo os dados cadastrais do jovem como: nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento do jovem, bem como, sua naturalidade e religião. Insiro ainda os nomes, profissões e telefones comerciais dos pais do jovem. Procuro no meu computador e seleciono uma foto deste escoteiro, registrada em sua inscrição. Seleciono a seção em que o novo escoteiro vai participar. Confirmo os dados que acabei de inserir e o sistema me retorna a confirmação de cadastro do escoteiro.

Cenário: C2.4 (Chefe) – Transferir um escoteiro da minha seção.

Contexto: Um escoteiro da minha seção está prestes a completar seus quinze anos e extremamente ansioso para participar da tropa sênior. Já a umas duas reuniões ele participa das atividades dos seniores e nesta ultima reunião ele fez sua passagem para a nova tropa, ou seja, para a nova fase de sua vida escoteira.

Objetivo: Alterar a seção de um escoteiro.

Ações: Eu acesso a área de gerenciamento e seleciono o jovem que irá fazer a passagem e informo a nova seção em que este escoteiro vai participar, confirmando minha ação.

Cenário: C2.5 (Chefe) – Desativar um escoteiro da minha seção.

Contexto: Um escoteiro de minha seção acaba de avisar que irá se mudar de cidade, desta forma, irá também se desligar do GE Ximbangue. Portanto devo desativá-lo do sistema para que ele não receba mais nenhuma atividade, conquista ou especialidade cadastradas a partir de agora.

Objetivo: Desativar um escoteiro de uma seção.

Ações: Eu acesso a área de gerenciamento e seleciono o jovem que desejo desativar confirmando minha ação.

Cenário: C2.6 (Chefe) – Registrar a data de uma conquista realizada por um escoteiro de minha seção.

Contexto: Sou chefe da seção escoteira, um escoteiro acaba de receber seu Lis de Ouro, para não passar em branco tenho que registrar a data desta conquista no esforçado jovem.

Objetivo: Registrar uma conquista a um escoteiro.

Ações: Eu acesso a área de gerenciamento e seleciono o jovem que desejo cadastrar a conquista. Eu escolho a conquista realizada pelo jovem em evidência, insiro a data em que esta foi realizada e o sistema retorna a Ficha 120 deste escoteiro com a data desta nova conquista registrada.

<p>Cenário: C2.7 (Chefe) – Registrar a data de uma progressão atingida por um escoteiro de minha seção.</p>
<p>Contexto: Sou chefe de uma alcatéia e um lobinho acaba de atingir o nível de progressão Saltador, para completar sua ficha 120 tenho que registrar a data desta progressão.</p>
<p>Objetivo: Registrar uma progressão a um escoteiro.</p>
<p>Ações: Eu acesso a área de gerenciamento e seleciono o jovem que desejo registrar a progressão. Eu escolho a progressão atingida pelo jovem em evidência, insiro a data em que esta foi realizada e o sistema retorna a Ficha 120 deste escoteiro com a data desta nova progressão registrada.</p>

<p>Cenário: C2.8 (Chefe) – Registrar a data e nível de uma especialidade realizada por um escoteiro de minha seção.</p>
<p>Contexto: Sou chefe da seção sênior e um escoteiro acaba de concluir seu nível dois na especialidade de cozinheiro, para completar sua ficha 120 tenho que registrar a data e o nível desta nova especialidade.</p>
<p>Objetivo: Registrar uma especialidade a um escoteiro.</p>
<p>Ações: Eu acesso a área de gerenciamento e seleciono o jovem que desejo registrar a nova especialidade. Eu seleciono área desta especialidade e o sistema informa as especialidades presentes nesta área, caso esta nova especialidade não esteja presente na relação informada pelo sistema, eu informo o nome da especialidade. Informo também a data em que esta foi realizada e os códigos referentes as etapas desta especialidade realizada pelo escoteiro. Então o sistema retorna a Ficha 120 deste escoteiro com a data, o nível e as etapas realizadas desta nova especialidade registrada.</p>

Cenário: C2.9 (Chefe) – Cadastrar uma atividade à minha seção.

Contexto: Sou chefe em uma seção do GE e minha seção realizou um acampamento neste último fim de semana. Então eu preciso registrar esta atividade a todos os escoteiros de minha seção presentes nesta atividade para que cada escoteiro possa acompanhar sua vida escoteira e apresentá-la aos seus pais.

Objetivo: Cadastrar uma atividade nos escoteiros de uma seção.

Ações: Eu acesso a área de gerenciamento de atividade. Escolho a opção de cadastrar uma nova atividade realizada. O sistema me fornece a relação das cinco últimas atividades realizadas por cada seção, do meu GE, com sua data, seu tipo, local e os ramos que participaram desta. Caso a atividade que eu quero cadastrar esteja presente nesta relação eu seleciono a atividade desejada pelo seu nome. Caso não esteja, eu escolho a opção de inserir uma nova atividade informando: os dias, mês e ano de sua realização, selecionando o tipo e o local pertinentes a ela, caso não exista o tipo ou local desejado, eu insiro um novo tipo ou local, e seleciono os ramos que participaram desta atividade. O sistema apresenta todos os escoteiros ativos na minha seção e eu seleciono os que participaram da atividade. Eu confirmo as informações apresentadas e então o sistema realiza o cadastro da nova atividade aos jovens da minha seção.

<p>Cenário: C2.10 (Chefe) – Ver a ficha 120 de um escoteiro.</p> <p>Contexto: Sou chefe do GE e desejo visualizar e imprimir a ficha 120 do escoteiro selecionado.</p> <p>Objetivo: Visualizar Ficha 120 de um escoteiro.</p>
<p>Ações: Eu selecio no o escoteiro pelo seu nome e o sistema apresenta os dados deste escoteiro no formato de ficha 120 com seu nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento, sua naturalidade e religião. Exibe ainda os nomes, profissões e telefones comerciais dos pais. A foto deste membro, a seção e o ramo as quais o membro pertence. A ficha ainda apresenta: o nome, a região, o numeral, o setor, o site e o e-mail do grupo escoteiro cadastrado no sistema; as datas de cada conquista atingida pelo escoteiro; as datas de cada progressão alcançada pelo escoteiro; a data e o nível de cada especialidade conquistada pelo jovem, bem como os índices da seqüência de tarefas que ele realizou; a data, o tipo, o local, os ramos e as observações de cada atividade realizada pelo escoteiro em sua vida escoteira. Eu ainda posso imprimir essas ficha ou voltar a relação de membros (Ver todos os escoteiros de sua seção) e selecionar outro escoteiro.</p>

<p>Cenário: C2.11 (Chefe) – Ver detalhes de um membro.</p> <p>Contexto: Estou visualizando a relação de membros adultos presentes no GE (Chefes ou Dirigentes) e desejo ver mais informações sobre este membro.</p> <p>Objetivo: Visualizar informações de um membro.</p>
<p>Ações: Eu selecio no o membro pelo nome e o sistema me retorna o nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento do membro, bem como, sua naturalidade e religião. Apresenta ainda sua foto, seu ramo e sua seção, caso pertença a uma.</p>

Cenário: C2.12 (Chefe) – Ver dirigentes presentes no GE.

Contexto: Sou chefe de uma seção do grupo escoteiro e preciso da relação de todos os dirigentes ativos no grupo, para agendar uma reunião.

Objetivo: Visualizar relação de dirigentes ativos existentes no GE.

Ações: Eu acesso a área de gerenciamento de dirigentes e o sistema me retorna a relação dos dirigentes presentes atualmente com seu registro, nome, telefone e e-mail, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um dirigente para ver mais informações (ver detalhes de um membro). Eu posso ainda retornar a lista e consultar os dados de outro membro.

Cenário: C2.13 (Chefe) – Alterar alguns campos dos meus dados cadastrais.

Contexto: Estou mudando de endereço e também de provedor de internet. Para que o sistema do GE continue atualizado, preciso alterar as minhas informações cadastrais.

Objetivo: Altera dados cadastrais de um membro.

Ações: Eu acesso a área de atualização de perfil de usuário. O sistema me retorna a relação de campos que eu posso alterar, em relação aos meus dados cadastrais como: endereço, bairro, cidade e CEP, telefone, celular, e-mail e religião. Eu escolho o campo que alterar e insiro o valor atualizado, após isso, confirmo minha alteração.

Cenário: C3.1 (Escoteiro) – Alterar alguns campos dos meus dados cadastrais.

Contexto: Estou mudando de endereço e também de provedor de internet. Para que o sistema do GE continue atualizado, preciso alterar as minhas informações cadastrais.

Objetivo: Altera dados cadastrais de um membro.

Ações: Eu acesso a área de atualização de perfil de usuário. O sistema me retorna a relação de campos que eu posso alterar, em relação aos meus dados cadastrais como: endereço, bairro, cidade e CEP, telefone, celular, e-mail e religião. Eu escolho o campo que alterar e insiro o valor atualizado, após isso, confirmo minha alteração.

Cenário: C3.2 (Escoteiro) – Ver todos os chefes de sua seção.

Contexto: Sou lobinho do GE Ximbangue e não poderei participar da reunião do próximo sábado. Meu chefe avisou que toda vez que eu for faltar tenho que avisa-lo previamente para justificar minha ausência na reunião. Para isto preciso saber o telefone, e-mail, celular ou o local onde meu chefe mora para poder entrar em contato.

Objetivo: Visualizar relação dos chefes ativos existentes em uma seção.

Ações: Eu acesso a área de Chefes da minha seção e o sistema me retorna a relação dos chefes presentes atualmente na minha seção com seu registro, nome, telefone e e-mail, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um chefe e ver mais detalhes sobre ele (Ver detalhes de um membro). Eu posso ainda retornar a lista e consultar os dados de outro chefe.

<p>Cenário: C3.3 (Escoteiro) – Ver todos os escoteiros de sua seção.</p>
<p>Contexto: Sou sênior do GE Ximbangue e meu chefe solicitou que eu reunisse todos os seniores da minha seção para preparar uma atividade para os lobinhos na próxima reunião, mas para isso preciso entrar em contato com o restante dos seniores da minha seção. Como sou novo na tropa não sei o endereço ou telefone de todos para poder agendar um horário e local para se reunir.</p>
<p>Objetivo: Visualizar relação dos escoteiros ativos existentes em uma seção.</p>
<p>Ações: Eu acesso a área de Escoteiros da minha seção e o sistema me retorna a relação dos escoteiros presentes atualmente na minha seção com seu registro, nome, telefone e e-mail, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um escoteiro e mais detalhes sobre ele (Ver detalhes de um membro). Eu posso ainda retornar a lista e consultar os dados de outro escoteiro.</p>

<p>Cenário: C3.4 (Escoteiro) – Ver detalhes de um membro.</p>
<p>Contexto: Estou visualizando a relação de membros presentes no GE (Escoteiros, chefes ou dirigentes) e desejo ver mais informações sobre este membro.</p>
<p>Objetivo: Visualizar informações de um membro.</p>
<p>Ações: Eu selecionei o membro pelo nome e o sistema me retorna o nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento do membro, bem como, sua naturalidade e religião. Apresenta ainda sua foto, seu ramo e sua seção, caso pertença a uma e os nomes, telefones e profissões de seus pais, caso seja um escoteiro.</p>

Cenário: C3.5 (Escoteiro) – Ver dirigentes presentes no GE.

Contexto: Sou escoteiro de uma seção do grupo escoteiro e preciso falar com um dirigentes ativos no grupo para acertar meu registro anual e minhas mensalidades.

Objetivo: Visualizar relação de dirigentes ativos existentes no GE.

Ações: Eu acesso a área de gerenciamento de dirigentes e o sistema me retorna a relação dos dirigentes presentes atualmente com seu registro, nome, telefone e e-mail, ordenados por seu nome. Eu posso imprimir essa relação ou selecionar um dirigente para ver mais informações (ver detalhes de um membro). Eu posso ainda retornar a lista e consultar os dados de outro membro.

Cenário: C3.6 (Escoteiro) – Visualizar e imprimir sua Ficha 120.

Contexto: Eu era escoteiro do GE Ximbangue e acabei de me mudar com minha família para o litoral. Onde irei ingressar em um novo grupo escoteiro daqui. Porém na correria da mudança esqueci de pegar minha Ficha 120, solicitada pelo meu novo chefe, com meu chefe do Ximbangue. Liguei para meu antigo chefe e ele me informou que eu poderia imprimir a Ficha 120 através do sistema escoteiro.

Objetivo: Visualizar Ficha 120 de um escoteiro.

Ações: Eu acesso a área de gerar minha Ficha 120 e o sistema apresenta os meus dados no formato de ficha 120 com meu nome, registro da UEB, o endereço completo, bairro, cidade e CEP, telefone, celular e e-mail para contato, a data de nascimento, minha naturalidade e religião. Exibe ainda os nomes, profissões e telefones comerciais dos meus pais. A minha foto, seção e o ramo as quais eu pertenço. A ficha ainda apresenta: o nome, a região, o numeral, o setor, o site e o e-mail do grupo escoteiro cadastrado no sistema; as datas de cada conquista atingida por mim; as datas de cada progressão alcançada por mim; a data e o nível de cada especialidade conquistada por mim, bem como os índices da seqüência de tarefas que eu já realizei com a data, o tipo, o local, os ramos e as observações de cada atividade realizada em minha vida escoteira. Eu ainda posso imprimir essas ficha.

9.2 RELAÇÃO DE USE CASES

Use Case 1: Alterar informações institucionais do GE

Cenários: C1.1

Descrição:

1. O usuário acessa a área de gerenciamento das informações cadastrais do GE.
2. O usuário informa nome, região, numeral, setor, cidade, estado, e-mail e URL atualizados do GE.
3. O usuário escolhe a opção de atualizar dados.

Use Case 2: Cadastrar Seção

Cenários: C1.2

Descrição:

1. O sistema apresenta a relação de seções presentes no GE com seus nomes, ramo e datas de criação.
2. O usuário seleciona a opção de cadastrar uma nova seção.
3. Informa o nome, ramo, data de criação e alguma observação, caso necessário, da nova seção.
4. O usuário escolhe a opção de inserir seção podendo retornar ao passo dois para inserir uma nova seção.

Use Case 3: Cadastrar membro

Cenários: C1.3/C1.9/C2.3

Descrição:

1. O usuário informa o nome, registro da UEB, o endereço completo, bairro, cidade, estado e CEP, telefone, celular e e-mail, data de nascimento, naturalidade, religião, nomes, profissão e telefone comercial dos pais, caso necessário.
2. Procura por uma foto do membro em seu computador.
3. Escolhe a seção, caso necessário, em que este membro participará e o tipo de membro que ele será (Chefe, dirigente ou escoteiro).
4. O usuário confirma seu cadastro.

Use Case 4: Ver relação de escoteiros por seção

Cenários: C1.4/ C2.5/ C2.1/ C3.3

Descrição:

7. O sistema apresenta o nome e ramo das seções presentes no grupo atualmente.
8. O usuário seleciona uma seção.
9. O sistema apresenta a relação de escoteiros presentes atualmente na seção com seu nome, registro na UEB, seção e ramo.
10. O usuário pode imprimir essa relação.
11. Caso o usuário seja um chefe ou dirigente ele pode ainda visualizar a ficha 120 do escoteiro, transferi-lo para outra seção ou desativá-lo, selecionando-o pelo nome.
12. Senão pode ver mais informações sobre o escoteiro selecionado (Ver detalhes sobre um membro).

Use Case 5: Ver ficha 120

Cenários: C1.5/ C2.10/ C3.6

Descrição:

1. O usuário seleciona o escoteiro pelo nome (Ver relação de escoteiros por seção)
2. O sistema apresenta o nome, a região, estado, o numeral, o setor, o site e o e-mail do grupo escoteiro; o registro na UEB, nome, endereço, bairro, cidade, CEP, telefone, celular e e-mail do membro, data e cidade de nascimento e religião, nomes, profissões e telefones comerciais dos pais e foto do membro. Apresenta ainda a data de cada conquista atingida pelo membro e a data de cada progressão alcançada. Assim como o nome, a data, o nível e tarefas de cada especialidade conquistada e a data, o tipo, o local, os ramos e as observações de cada atividade realizada pelo membro em sua vida escoteira.
3. O usuário pode ainda imprimir essas informações.

Use Case 6: Ver membros por seção

Cenários: C1.6/ C1.10/ C2.2/ C2.12/ C3.2/ C3.5

Descrição:

1. O sistema apresenta o nome e ramo e data de criação das seções presentes no grupo atualmente.
2. O usuário seleciona uma seção.
3. O sistema apresenta a relação de membros presentes atualmente na seção com seu nome, registro na UEB, telefone e e-mail.
4. O usuário pode imprimir essa relação.
5. Caso o usuário seja um dirigente ele pode ainda transferir este chefe para outra seção(Transferir um chefe de seção) ou desativá-lo.
6. O usuário pode ver mais informações sobre o membro selecionando-o pelo nome (Ver detalhes sobre um membro).

Use Case 7: Transferindo um membro para uma seção

Cenários: C1.7/ C1.8/C2.4

Descrição:

1. O usuário seleciona um membro pelo seu nome e escolhe a nova seção em que este membro irá participar.
2. O usuário confirma a operação.

Use Case 8: Registrar datas significativas

Cenários: C2.6/ C2.7/ C2.8

Descrição:

1. O usuário seleciona o escoteiro (Ver relação de escoteiros por seção) que deseja editar.
2. Caso seja uma especialidade o usuário seleciona a área da especialidade.
3. O usuário informa a data, o nível e a conquista, especialidade ou progressão atingida.
4. Caso esta não esteja registrada, o usuário informa o nome do evento.
5. Se esta for uma especialidade, o usuário pode informar as tarefas completadas.
6. O usuário confirma e voltar ao passo um podendo escolher outro membro.

Use Case 9: Cadastrar Atividade

Cenários: C2.9

Descrição:

1. O sistema apresenta a relação das cinco ultimas atividades realizadas por cada seção do sistema com seu tipo, data, local e ramos.
2. O usuário escolhe a atividade desejada.
3. Caso a atividade não esteja presente na relação o usuário informa os dias, mês e ano, o tipo, local e ramos que participaram dessa atividade.
4. Caso não estejam presentes o local ou tipo de atividade desejados, o usuário pode informar estes novos dados.
5. O sistema apresenta todos os escoteiros ativos da seção do usuário.
6. O usuário seleciona os que participaram da atividade.
7. O usuário confirma o cadastro e pode voltar ao passo um.

Use Case 10: Ver detalhes de um membro

Cenários: C1.11/ C2.11/ C3.4

Descrição:

1. O sistema apresenta o registro na UEB, nome, endereço, bairro, cidade, CEP, telefone, celular e e-mail do membro, data e cidade de nascimento e religião, nomes, profissões e telefones comerciais dos pais, caso necessário, e foto do membro.

Use Case 11: Alterar dados pessoais

Cenários: C1.12/ C2.13 /C3.1

Descrição:

1. O usuário informa as informações que pode alterar com seu: endereço, bairro, cidade e CEP, telefone, celular, e-mail e religião. Profissão e telefones comerciais dos pais, caso necessário.
2. O usuário confirma com a opção de atualizar dados.

9.3 RELAÇÃO DE UIDS

UID 01: Alterar informações institucionais do GE

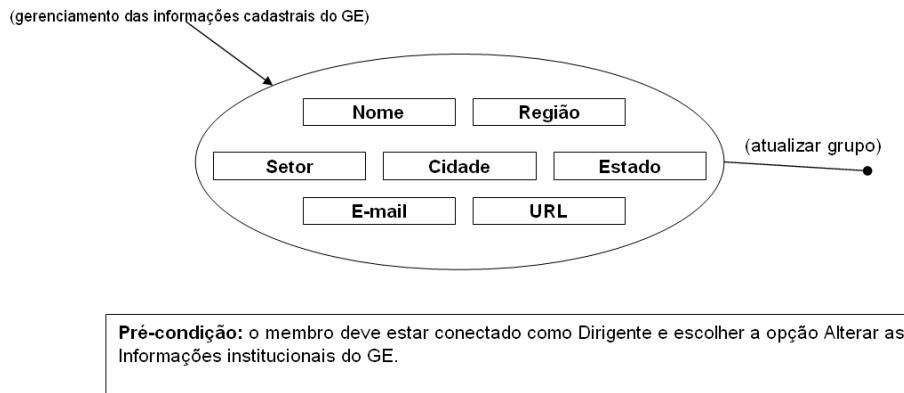


Figura 52: UID 01 Alterar informações institucionais do GE.

UID 02: Cadastrar Seção

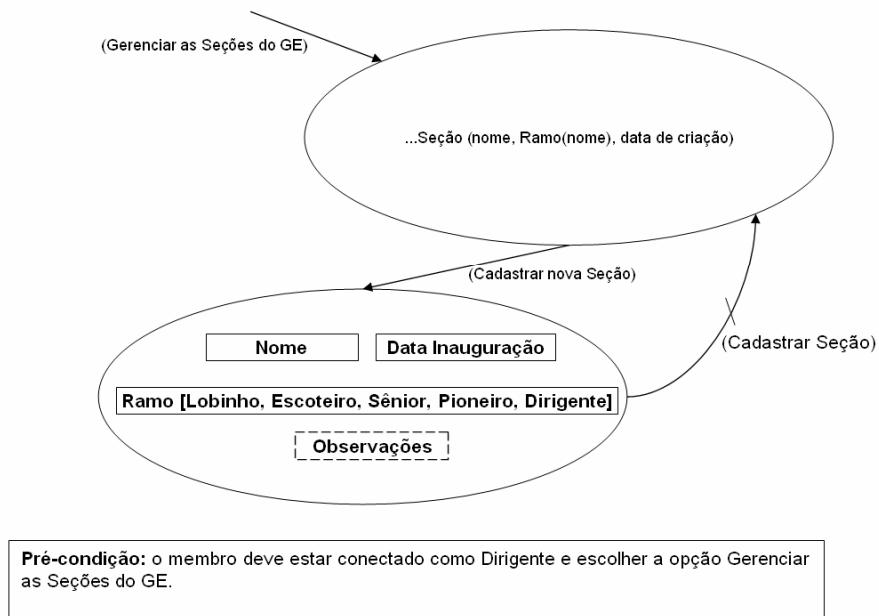


Figura 53: UID 02 Cadastrar Seção.

UID 03: Cadastrar Membro

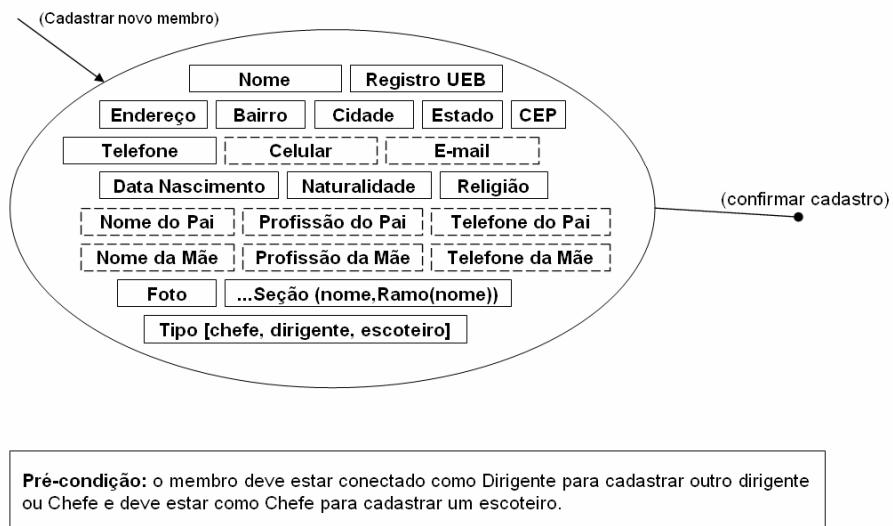
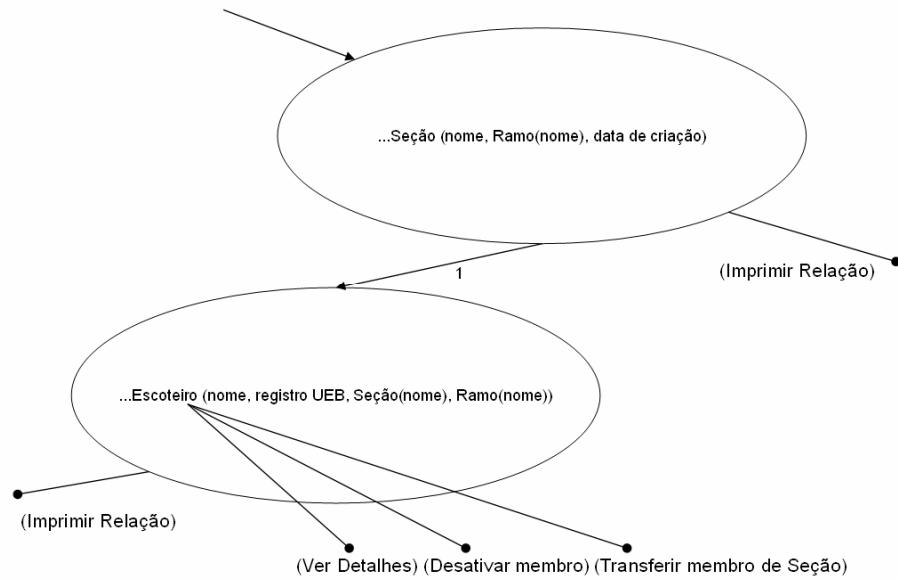


Figura 54: UID 03 Cadastrar Membro.

UID 04: Ver relação de escoteiros por seção



Pré-condição: caso o membro esteja cadastrado como Dirigente ou Chefe, “Ver Detalhes” mostrará a ficha 120 do escoteiro selecionado e estarão ativas as opções de transferir membro de seção e desativar membro.

Figura 55: UID Ver relação de escoteiros por seção.

UID 05: Ver Ficha 120

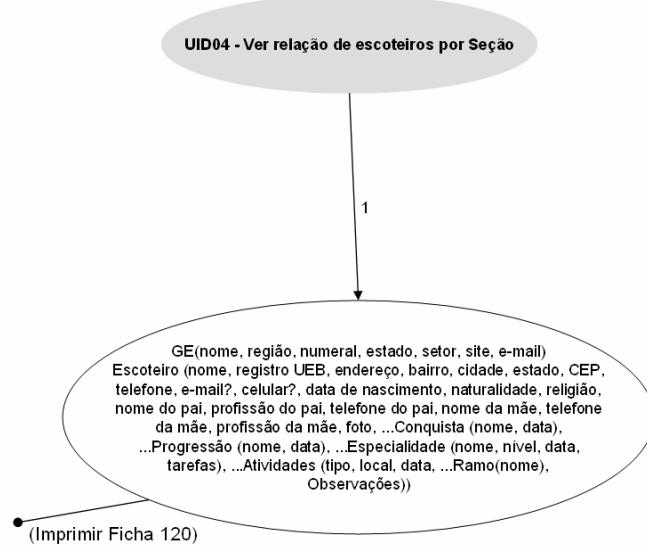
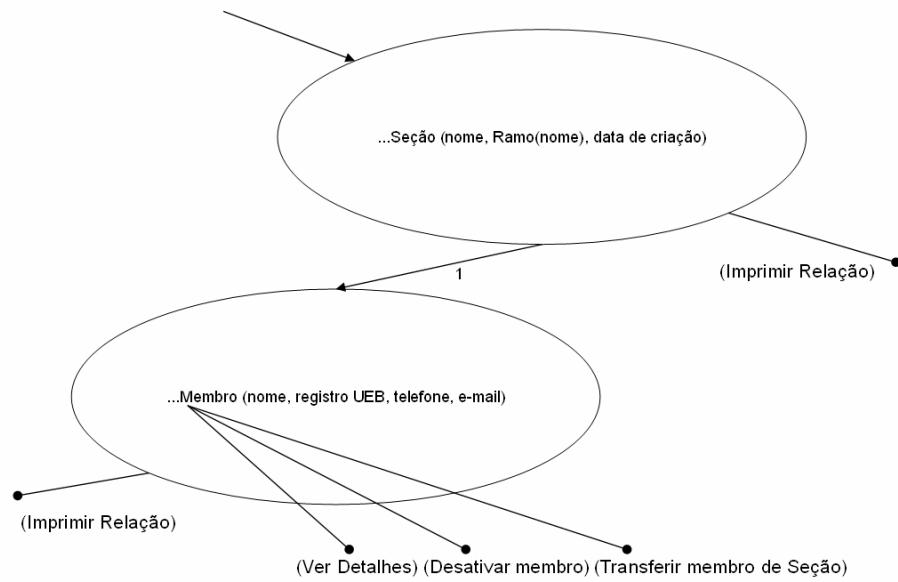
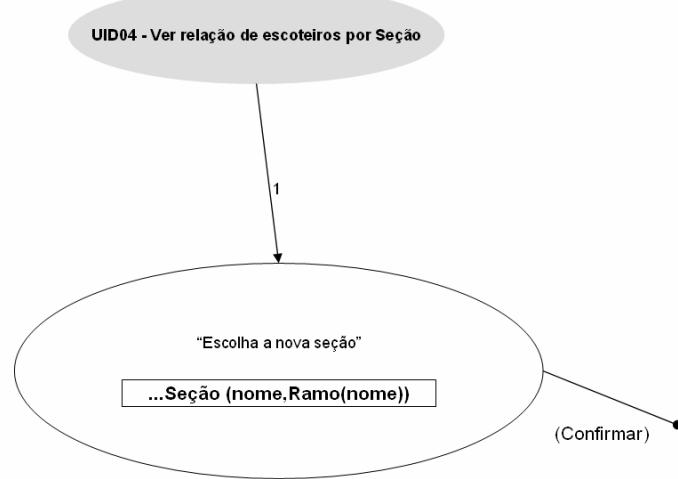


Figura 56: UID 05 Ver ficha 120.

UID 06: Ver membros por seção

Pré-condição: caso o membro esteja cadastrado como Dirigente estarão ativas as opções de transferir membro de seção e desativar membro.

Figura 57: UID 06 Ver membros por seção.

UID 07: Transferir um membro para uma seção

Pré-condição: ter sido escolhido no UID04 a opção de transferir um membro para outra seção.

Figura 58: UID 07 Transferir um membro para uma seção.

UID 08: Registrar datas significativas

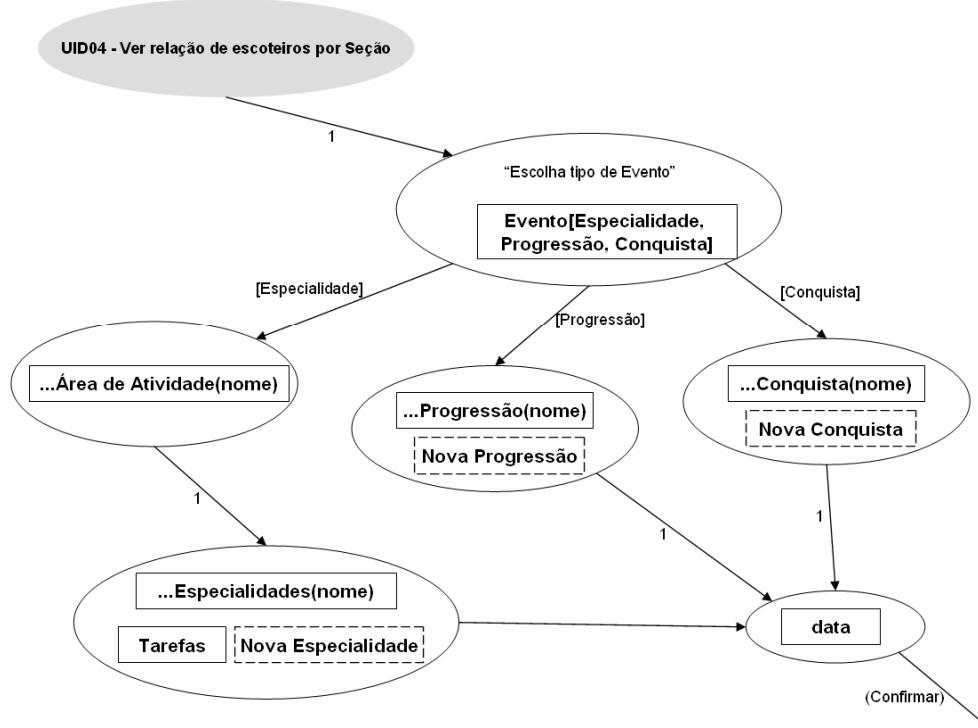


Figura 59: IUD 08 Registrar datas significativas.

UID 09: Registrar Atividades

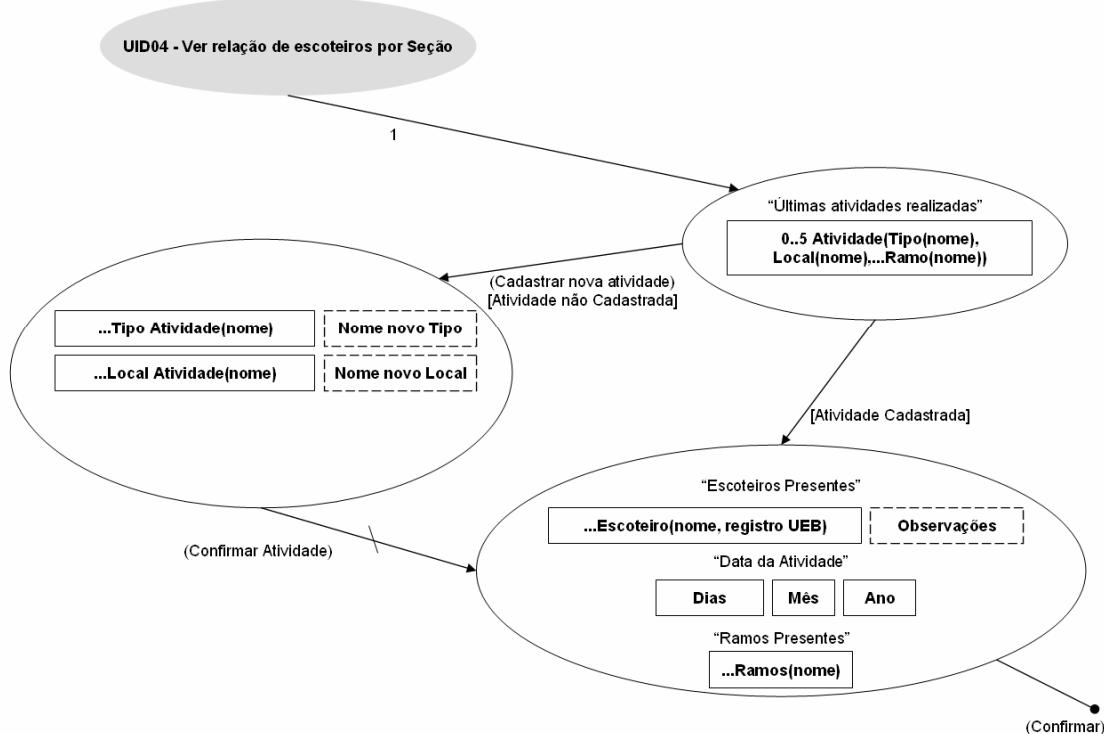


Figura 60: UID 09 Registrar Atividades.

UID 10: Ver detalhes de um membro

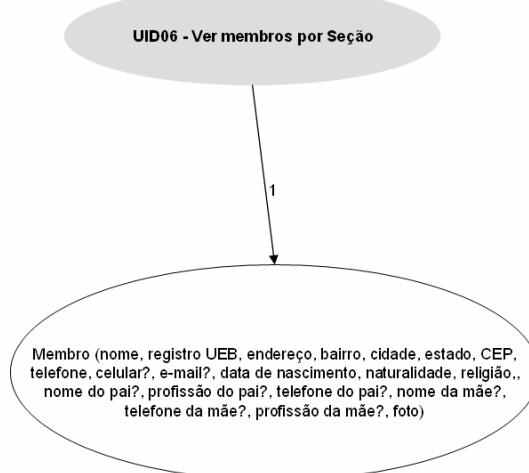


Figura 61: UID 10 Ver detalhes de um membro.

UID 11: Alterar dados pessoais

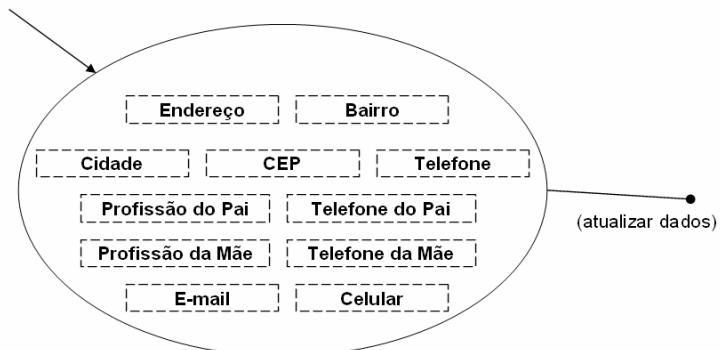


Figura 62: UID 11 Alterar dados pessoais.

9.4 RELAÇÃO DE CARTÕES DE VISÃO

9.4.1 Ator: DIRIGENTE

Membro Dirigente Alfabético
Membro Chefe por Seção
Membro Escoteiro por Seção

nome, registro UEB, endereço, bairro, cidade, estado, CEP, telefone, celular?, e-mail?, data de nascimento, naturalidade, religião, nome do pai?, profissão do pai?, telefone do pai?, nome da mãe?, telefone da mãe?, profissão da mãe?, foto

me-seção: se.nome where me pertence se
inf-seção: anchor(Ctx Seções Alfabético (se:Seção where me participa se))

alterarMeusDados()

Dirigente Leitura

Membro Dirigente Criação
Membro Chefe Criação

addMembro()

Dirigente Escrita

Membro Dirigente Alteração

setMembro(self)

Dirigente Escrita

Membro Chefe Alterar Seção

setChefe()

Dirigente Escrita

GE

nome, região, setor, cidade, estado, email, url

seções: Idx Seções Alfabético (self)
dirigentes: Idx Dirigente Alfabético (self)

`setGE()`

Dirigente Escrita

Seções Alfabético

nome, criação, observações

ramo: r:Ramo, r.nome where s:Seção pertence_a r

chefes: Idx Chefe por Seção (self)

escoteiros: Idx Escoteiro por Seção (self)

imprimirSeção()

Dirigente Leitura

Seções Criação

addSeção()

Dirigente Escrita

Seções Alteração

`setSeção()`

Dirigente Escrita

9.4.2 Ator: CHEFE

Membro Dirigente Alfabético

Membro Chefe por Seção

Membro Escoteiro por Seção

Membro Escoteiro por Atividade

nome, registro UEB, endereço, bairro, cidade, estado, CEP, telefone, celular?, e-mail?, data de nascimento, naturalidade, religião, nome do pai?, profissão do pai?, telefone do pai?, nome da mãe?, telefone da mãe?, profissão da mãe?, foto

alterarMeusDados()

Dirigente Leitura

Membro Escoteiro Criação

addEscoteiro()

Chefe Escrita

Membro Alteração

setMembro(self)

Chefe Escrita

Atividades por Seção

Atividades por Escoteiro

observação, dias, mês, ano

from at:Atividade, la:LocalAtividade, ta:TipoAtividade

ramos: list of <ra:Ramo, ra.nome where ra participa at>

local: la.nome where at acontece la

tipo: ta.nome where at foi ta

escoteiros: Idx Escoteiro por Seção (self)

cadastrarAtividade()

Chefe Escrita

Evento Criação

addEvento()

Chefe Escrita**Ficha 120 por Escoteiro**

```
from ge:GE, es:Escoteiro, ct:Conquista, pg:Progressão, ep:Especialidade, fc:Ficha120
ge.nome, ge.região, ge.numeral, ge.setor, ge.url, ge.e-mail,
es.registro, es.nome, es.endereço, es.bairro, es.cidade, es.estado, es.CEP,
es.telefone, es.celular, es.e-mail, es.data nascimento, es.naturalidade, es.religião,
es.pai, es.profissão pai, es.telefone pai, es.mãe, es.profissão mãe, es.telefone mãe, es.foto
```

prog-name:	List of<pg.nome where fc tem pg>
prog-data:	List of<pg.data where fc tem pg>
conq-name:	List of<ct.nome where fc tem ct>
conq-data:	List of<ct.data where fc tem ct>
esp-name:	List of<ep.nome where fc tem ep>
esp-data:	List of<ep.data where fc tem ep>
esp-nível:	List of<ep.nível where fc tem ep>
esp-área:	List of<ep.área where fc tem ep>
esp-tarefas:	List of<ep.tarefas where fc tem ep>
atividades:	Idx Atividades por Escoteiro (es)

imprimirFicha()

Chefe Escrita**GE**

nome, região, setor, cidade, estado, email, url

seções:	Idx Seções Alfabético (self)
dirigentes:	Idx Dirigente Alfabético (self)

Chefe Leitura

9.4.3 Ator: ESCOTEIRO

Membro Dirigente Alfabético

Membro Chefe por Seção

Membro Escoteiro por Seção

Membro Escoteiro por Atividade

nome, registro UEB, endereço, bairro, cidade, estado, CEP, telefone, celular?, e-mail?, data de nascimento, naturalidade, religião, nome do pai?, profissão do pai?, telefone do pai?, nome da mãe?, telefone da mãe?, profissão da mãe?, foto

alterarMeusDados()

Escoteiro Escrita

Membro Alteração

setMembro(self)

Escoteiro Escrita

Atividades por Seção

Atividades por Escoteiro

observação, dias, mês, ano

from at:Atividade, la:LocalAtividade, ta:TipoAtividade

ramos: list of <ra:Ramo, ra.nome where ra participa at>

local: la.nome where at acontece la

tipo: ta.nome where at foi ta

escoteiros: Idx Escoteiro por Seção (self)

Escoteiro Leitura

Ficha 120 por Escoteiro

from ge:GE, es:Escoteiro, ct:Conquista, pg:Progressão, ep:Especialidade, fc:Ficha120
 ge.nome, ge.região, ge.numeral, ge.setor, ge.url, ge.e-mail,
 es регистрация, es.nome, es.endereço, es.bairro, es.cidade, es.estado, es.CEP,
 es.telefone, es.celular, es.e-mail, es.data_nascimento, es.naturalidade, es.religião,
 es.pai, es.profissão_pai, es.telefone_pai, es.mãe, es.profissão_mãe, es.telefone_mãe, es.foto

prog-nome: List of<pg.nome where fc tem pg>
 prog-data: List of<pg.data where fc tem pg>
 conq-nome: List of<ct.nome where fc tem ct>
 conq-data: List of<ct.data where fc tem ct>
 esp-nome: List of<ep.nome where fc tem ep>
 esp-data: List of<ep.data where fc tem ep>
 esp-nível: List of<ep.nível where fc tem ep>
 esp-área: List of<ep.área where fc tem ep>
 esp-tarefas: List of<ep.tarefas where fc tem ep>
 atividades: Idx Atividades por Escoteiro (es)

imprimirFicha()

Escoteiro Leitura

GE

nome, região, setor, cidade, estado, email, url

dirigentes: Idx Dirigente Alfabético (self)

Escoteiro Leitura

9.5 RELAÇÃO DE CARTÕES DE ESPECIFICAÇÃO

9.5.1 Cartões de Contexto

Contexto: Dirigente Alfabético			
Parâmetros			
Elementos: d:Dirigente			
Classe em Contexto:			
Ordenação por d.nome, ascendente			
Navegação Interna: por índice (Idx Dirigente Alfabético(d))			
Operações			
Usuário:	Dirigente	Permissão:	Leitura
	Chefe		Leitura
	Escoteiro		Leitura
Comentários:			

Contexto: Dirigente Criação			
Parâmetros			
Elementos: d:Dirigente			
Classe em Contexto: Membro Dirigente			
Ordenação por d.nome, ascendente			
Navegação Interna: por índice (Idx Dirigente Alfabético(d))			
Operações			
Usuário:	Dirigente	Permissão:	Escrita
Comentários:			

Contexto: Chefe por Seção			
Parâmetros			
Elementos: c:Chefe where c.atua_em s:Seção			
Classe em Contexto:			
Ordenação por c.nome, ascendente			
Navegação Interna: por índice (Idx Chefe por Seção)			
Operações			
Usuário:	Dirigente	Permissão:	Leitura
	Chefe		Leitura
	Escoteiro		Leitura
Comentários:			

Contexto: Chefe Criação			
Parâmetros			
Elementos: c:Chefe			
Classe em Contexto: Membro Chefe			
Ordenação por c.nome, ascendente			
Navegação Interna: por índice (Idx Chefe por Seção)			
Operações			
Usuário:	Dirigente	Permissão:	Escrita
Comentários:			

Contexto: Escoteiro por Seção			
Parâmetros			
Elementos: e:Escoteiro where e participa s:Seção			
Classe em Contexto:			
Ordenação por e.nome, ascendente			
Navegação Interna: por índice (Idx Escoteiro por Seção)			
Operações			
Usuário:	Dirigente	Permissão:	Leitura
	Chefe		Leitura
	Escoteiro		Leitura
Comentários:			

Contexto: Escoteiro por Atividade			
Parâmetros			
Elementos: e:Escoteiro where e participa a:Atividade			
Classe em Contexto:			
Ordenação por e.nome, ascendente			
Navegação Interna: por índice (Idx Escoteiro por Atividade)			
Operações			
Usuário:	Chefe	Permissão:	Leitura
	Escoteiro		Leitura
Comentários:			

Contexto: Escoteiro Criação			
Parâmetros			
Elementos: c:Escoteiro			
Classe em Contexto: Membro Escoteiro			
Ordenação por e.nome, ascendente			
Navegação Interna: por índice (Idx Escoteiro por Seção)			
Operações			
Usuário:	Chefe	Permissão:	Escrita
	Escoteiro		
Comentários:			

Contexto: Membro Alteração			
Parâmetros			
Elementos: m:Membro			
Classe em Contexto: Membro			
Ordenação por m.nome, ascendente			
Navegação Interna: por índice (Idx Escoteiro por Seção), (Idx Chefe por Seção), (Idx Dirigente Alfabético)			
Operações			
Usuário:	Chefe	Permissão:	Escrita
	Escoteiro		
Comentários:			

Contexto: Atividade por Seção			
Parâmetros			
Elementos: a:Atividade where e:Escoteiro participa a AND e participa s			
Classe em Contexto:			
Ordenação por a.dia*, a.mês, a.ano, ascendente			
Navegação Interna: por índice (Idx Atividade por Seção)			
Operações			
Usuário:	Chefe	Permissão:	Leitura
	Escoteiro		Leitura
Comentários:			

Contexto: Atividade por Escoteiro			
Parâmetros			
Elementos: a:Atividade where e:Escoteiro participa a			
Classe em Contexto:			
Ordenação por a.dia*, a.mês, a.ano, ascendente			
Navegação Interna: por índice (Idx Atividade por Escoteiro)			
Operações			
Usuário:	Chefe	Permissão:	Leitura
	Escoteiro		Leitura
Comentários:			

Contexto: Atividade Criação			
Parâmetros			
Elementos: a:Atividade where e:Escoteiro participa a AND e participa s			
Classe em Contexto: Atividade Criação			
Ordenação por a.dia*, a.mês, a.ano, ascendente			
Navegação Interna: por índice (Idx Atividade por Seção)			
Operações			
Usuário:	Chefe	Permissão:	Escrita
Comentários:			

Contexto: GE Criação			
Parâmetros			
Elementos: ge:GE			
Classe em Contexto: GE			
Ordenação			
Navegação Interna:			
Operações			
Usuário:	Dirigente	Permissão:	Escrita
Comentários:			

Contexto: Evento por Escoteiro			
Parâmetros			
Elementos: e:Evento where es:Escoteiro es pertence fc:Ficha120 AND fc tem e			
Classe em Contexto:			
Ordenação por e.data*, e.nome, ascendente			
Navegação Interna: por anchor (Ctx Ficha120 por Escoteiro)			
Operações			
Usuário:	Chefe	Permissão:	Leitura
Comentários:			

Contexto: Evento Criação			
Parâmetros			
Elementos: e:Evento where es:Escoteiro es pertence fc:Ficha120 AND fc tem e			
Classe em Contexto: Evento			
Ordenação por e.data*, e.nome, ascendente			
Navegação Interna: por anchor (Ctx Ficha120 por Escoteiro)			
Operações			
Usuário:	Chefe	Permissão:	Escrita
Comentários:			

Contexto: Ficha120 por Escoteiro			
Parâmetros			
Elementos: fc:Ficha120, ge:GE, es:Escoteiro, ev:Eventos, at:Atividades where fc pertence es AND fc possui ge AND fc tem ev AND fc tem at			
Classe em Contexto:			
Ordenação			
Navegação Interna: por anchor (Ctx Ficha120 por Escoteiro)			
Operações			
Usuário:	Chefe Escoteiro	Permissão:	Leitura Leitura
Comentários: O escoteiro tem permissão de ver somente a sua ficha 120 e o chefe as fichas de somente sua seção.			

Contexto: Seção Alfabético			
Parâmetros			
Elementos: sc:Seção			
Classe em Contexto:			
Ordenação			
Navegação Interna: por sc.nome, ascendente			
Operações			
Usuário:	Dirigente Chefe Escoteiro	Permissão:	Todas Somente a Sua Somente a Sua
Comentários:			

Contexto: Seção Criação			
Parâmetros			
Elementos: sc:Seção			
Classe em Contexto: Seção			
Ordenação			
Navegação Interna: por sc.nome, ascendente			
Operações			
Usuário:	Dirigente	Permissão:	Escrita
Comentários: Nova seção			

Contexto: Seção Alteração			
Parâmetros			
Elementos: sc:Seção			
Classe em Contexto: Seção			
Ordenação			
Navegação Interna: por sc.nome, ascendente			
Operações			
Usuário:	Dirigente	Permissão:	Escrita
Comentários: Desativar e reativar seção			

9.5.2 Cartões de Estrutura de Acesso

Estrutura de Acesso: GE: Dirigentes			
Nível			
GE	Elementos: Dom (GE)		
	Atributos	Destino	
	ge: GE, ge.nome, ge.região, ge.setor, ge.cidade, ge.estado, ge.email, ge.url	Idx Dirigentes Alfabético (ge.nome)	
	Ordenação: por ge.nome, ascendente		
Dirigentes por GE	Parâmetros: ge: GE		
	Elementos: c: Chefe where c.ge = ge		
	Atributos	Destino	
	d: Dirigentes, d.nome, d регистра, d.telefone, d.email	Ctx Dirigentes Alfabético (d.registro)	
	Ordenação: por d.nome, ascendente		
	Usuário:	Permissão:	
	Dirigente	Leitura	
	Chefes	Leitura	
	Escoteiros	Leitura	
Comentários:			
Depende de:		Influencia:	

Estrutura de Acesso: GE: Seções			
Nível			
GE	Elementos: Dom (GE)		
	Atributos		
	ge: GE, ge.nome, ge.região, ge.setor, ge.cidade, ge.estado, ge.email, ge.url		
	Ordenação: por ge.nome, ascendente		
Seções por GE	Parâmetros: ge: GE		
	Elementos: s: Seção where s.ge = ge		
	Atributos	Destino	
	s: Seção, s.nome, s.criação, s.observação, s.ramo	Ctx Seções Alfabético (s.nome)	
	Ordenação: por d.nome, ascendente		
	Usuário:	Permissão:	
	Dirigente	Leitura	
Comentários:			
Depende de:		Influencia:	

Estrutura de Acesso: Seção: Chefs		
Nível		
Seção	Elementos: Dom (Seção) Atributos s: Seção, s.nome, s.criação, s.observação, s.ramo Ordenação: por s.nome, ascendente	
	Parâmetros: s: Seção	
	Elementos: c: Chefe where c.seção = s Atributos c: Chefe, c.nome, c.registro, c.e-mail, c.telefone Ordenação: por c.nome, ascendente	Destino Idx Chefs por Seção (s.nome) Destino Ctx Chefs por Seção (c.nome)
Usuário:	Dirigente Chefe Escoteiro	Permissão: Leitura Leitura Leitura
Comentários:		
Depende de:		Influencia:

Estrutura de Acesso: Seção: Escoteiros		
Nível		
Seção	Elementos: Dom (Seção) Atributos s: Seção, s.nome, s.criação, s.observação, s.ramo Ordenação: por s.nome, ascendente	
	Parâmetros: s: Seção	
	Elementos: e: Escoteiro where e.seção = s Atributos e: Escoteiro, e.nome, e.registro, e.e-mail, e.telefone Ordenação: por e.nome, ascendente	Destino Idx Escoteiros por Seção (s.nome) Destino Ctx Escoteiros por Seção (e.nome)
Usuário:	Dirigente Chefe Escoteiro	Permissão: Leitura Leitura Leitura
Comentários:		
Depende de:		Influencia:

Estrutura de Acesso: Seção: Atividades			
Nível			
Seção	Elementos: Dom (Seção)		
	Atributos s: Seção, s.nome, s.criação, s.observação, s.ramo		Destino Idx Atividades por Seção (s.nome)
	Ordenação: por s.nome, ascendente		
Atividades por Seção	Parâmetros: s: Seção		
	Elementos: a: Atividade where es:Escoteiro participa a AND es participa seção		
	Atributos a: Atividade, a.observação, a.dias, a.mês, a.ano, a.local, a.tipo, a.ramo		Destino Ctx Atividades por Seção (a)
	Ordenação: por a.dias, a.mês, a.ano, a.local, ascendente		
Usuário:	Chefe Escoteiro	Permissão:	Leitura Leitura
Comentários:			
Depende de:	Influencia:		

Estrutura de Acesso: Minhas Atividades			
Parâmetros: e: Escoteiro			
Elementos: a: Atividade where e participa a			
Atributos a: Atividade, a.observação, a.dias, a.mês, a.ano, a.local, a.tipo, a.ramo		Destino Ctx Atividades por Escoteiro (a)	
Ordenação: por a.dias, a.mês, a.ano, a.local, ascendente			
Usuários: Escoteiro	Permissões: Leitura		
Comentários:			
Depende de:	Influencia:		

Estrutura de Acesso: Minha Ficha 120			
Parâmetros: e: Escoteiro			
Elementos: a: Atividade where e participa a			
Atributos a: Atividade, a.observação, a.dias, a.mês, a.ano, a.local, a.tipo, a.ramo		Destino Ctx Atividades por Escoteiro (a)	
Ordenação: por a.dias, a.mês, a.ano, a.local, ascendente			
Usuários: Escoteiro	Permissões: Leitura		
Comentários:			
Depende de:	Influencia:		

9.6 RELAÇÃO DE ADVS



Figura 63: ADV de Estrutura da WebApp



Figura 64: ADVs de Conteúdo da WebApp

ADV Editar Registro	ADV Editar Naturalidade	ADV Editar Nome	ADV Editar Nome
Registro: String em Edição	Naturalidade: String em Edição	Nome: String em Edição	Setor: String em Edição
ADV Editar Foto	ADV Editar Religião	ADV Editar Região	ADV Editar Criação
Foto: Imagem em Edição	Religião: String em Edição	Região: String em Edição	Criação: Data em Edição
ADV Editar Endereço	ADV Editar Pai-nome	ADV Editar Setor	ADV Editar Observação
Endereço: String em Edição	Pai-nome: String em Edição	Setor: String em Edição	Observação: Texto em Edição
ADV Editar Bairro	ADV Editar Pai-profissão	ADV Editar Cidade	ADV Editar Dias
Bairro: String em Edição	Pai-profissão: String em Edição	Cidade: String em Edição	Dias: String em Edição
ADV Editar CEP	ADV Editar Pai-telefone	ADV Editar Estado	ADV Editar Mês
CEP: Inteiro em Edição	Pai-telefone: Inteiro em Edição	Estado: String em Edição	Mês: lista de (Meses do Ano)
ADV Editar Telefone	ADV Editar Mãe-nome	ADV Editar E-mail	ADV Editar Ano
Telefone: Inteiro em Edição	Mãe-nome: String em Edição	E-mail: String em Edição	Ano: Inteiro em Edição
ADV Editar Celular	ADV Editar Mãe-profissão	ADV Editar URL	ADV Editar Local
Celular: Inteiro em Edição	Mãe-profissão: String em Edição	URL: Texto em Edição	Local: lista de (LocaAtividade)
ADV Editar Data Nascimento	ADV Editar Mãe-telefone	ADV Alterar Dados	ADV Editar Tipo
DataNasc: Data em Edição	Mãe-telefone: Inteiro em Edição	É Botão	Tipo: lista de (TipoAtividade)
ADV Editar Senha	ADV Editar Lista Escoteiros		
Senha: Senha em Edição	Escoteiros: lista de (Escoteiros por Seção)		
ADV Editar Ramo			
	Ramo: lista de [Lobinho, Escoteiro, Sênior, Pioneiro, Chefia] em Edição		

Figura 65: ADVs atômicos da WebApp.

ADV Autenticação
ADV Edição Registro ADV Edição Senha ADV Confirmar

Figura 66: ADV de Autenticação de Usuário

ADV GE	ADV Atualização de GE
Nome: String Região: String Setor: String Cidade: String Estado: String e-mail: âncora (self e-mail) URL: âncora (self URL) ADV Índice de Seção Ativa ADV Índice de Dirigente Ativo	ADV Editar Nome ADV Editar Região ADV Editar Setor ADV Editar Cidade ADV Editar Estado ADV Editar E-mail ADV Editar URL ADV Confirmar ADV Cancelar

Figura 67: ADVs de Visualização e Edição de GE.

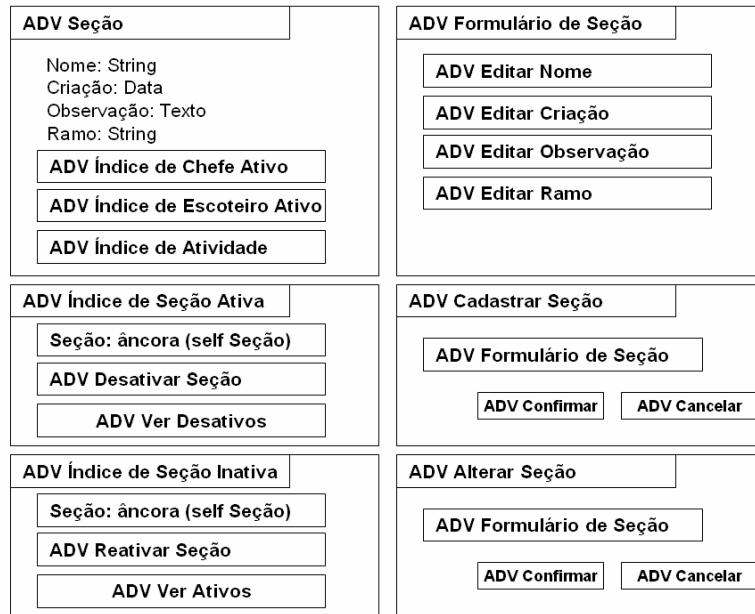


Figura 68: ADVs de Visualização, Edição e Listagem de Seções.

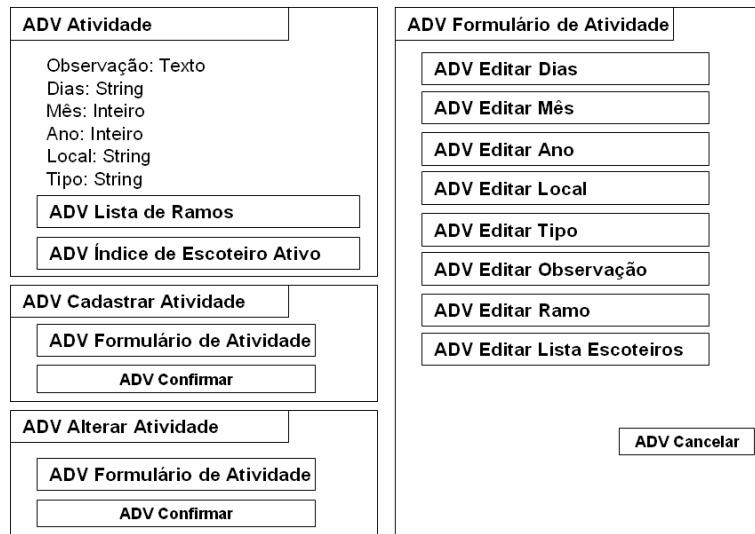


Figura 69: ADVs de Visualização e Edição de Seções.

ADV Membro	ADV Membro
Registro: Inteiro	ADV Editar Registro
Nome: String	ADV Editar Senha
Foto: Imagem	ADV Editar Nome
Endereço: String	ADV Editar Foto
Bairro: String	ADV Editar Endereço
Cidade: String	ADV Editar Bairro
Estado: String	ADV Editar Cidade
CEP: Inteiro	ADV Editar Estado
Telefone: Inteiro	ADV Editar CEP
Celular: Inteiro	ADV Editar E-mail
E-mail: String	ADV Editar Telefone
Data-nascimento: Data	ADV Editar Celular
Naturalidade: String	ADV Editar Data Nascimento
Religião: String	ADV Editar Naturalidade
Pai-name: String	ADV Editar Pai-nome
Pai-profissão: String	ADV Editar Mãe-nome
Pai-telefone: Inteiro	ADV Editar Pai-profissão
Mãe-name: String	ADV Editar Mãe-profissão
Mãe-profissão: String	ADV Editar Pai-telefone
Mãe-telefone: Inteiro	ADV Editar Mãe-telefone
	ADV Editar Religião
	ADV Cancelar

Figura 70: ADVs de Visualização e Edição de Membros.

ADV Cadastrar Dirigente	ADV Cadastrar Chefe	ADV Cadastrar Escoteiro
ADV Formulário Membro	ADV Formulário Membro	ADV Formulário Membro
ADV Confirmar	ADV Editar Seção	ADV Editar Seção
	ADV Confirmar	ADV Confirmar
ADV Editar Dirigente	ADV Editar Chefe	ADV Editar Escoteiro
ADV Formulário Membro	ADV Formulário Membro	ADV Formulário Membro
ADV Confirmar	ADV Editar Seção	ADV Editar Seção
	ADV Confirmar	ADV Confirmar
ADV Índice Dirigente Ativo	ADV Índice Chefe Ativo	ADV Índice Escoteiro Ativo
Dirigente: âncora (self Dirigente)	Chefe: âncora (self Chefe)	Escoteiro: âncora (self Escoteiro)
ADV Desativar Dirigente	ADV Desativar Chefe	ADV Desativar Escoteiro
ADV Ver Desativos	ADV Ver Desativos	ADV Ver Desativos
ADV Índice Dirigente Inativo	ADV Índice Chefe Inativo	ADV Índice Escoteiro Inativo
Dirigente: âncora (self Dirigente)	Chefe: âncora (self Chefe)	Escoteiro: âncora (self Escoteiro)
ADV Reativar Dirigente	ADV Reativar Chefe	ADV Reativar Escoteiro
ADV Ver Ativos	ADV Ver Ativos	ADV Ver Ativos

Figura 71: ADVs de Criação, Edição e Listagem de Membros.

9.7 RELAÇÃO DE DIAGRAMAS DE CONFIGURAÇÃO

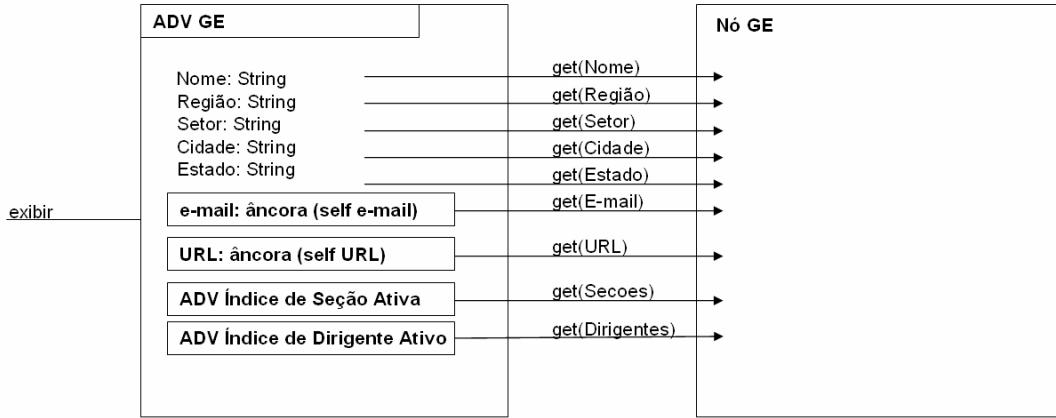


Figura 72: Diagrama de configuração do ADO GE.

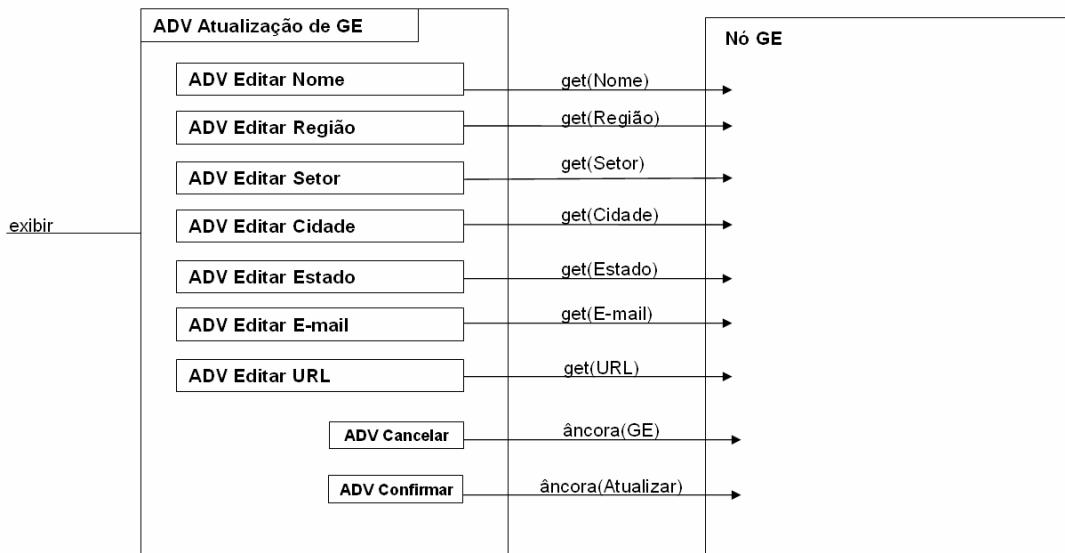


Figura 73: Diagrama de configuração do ADO Formulário de GE.

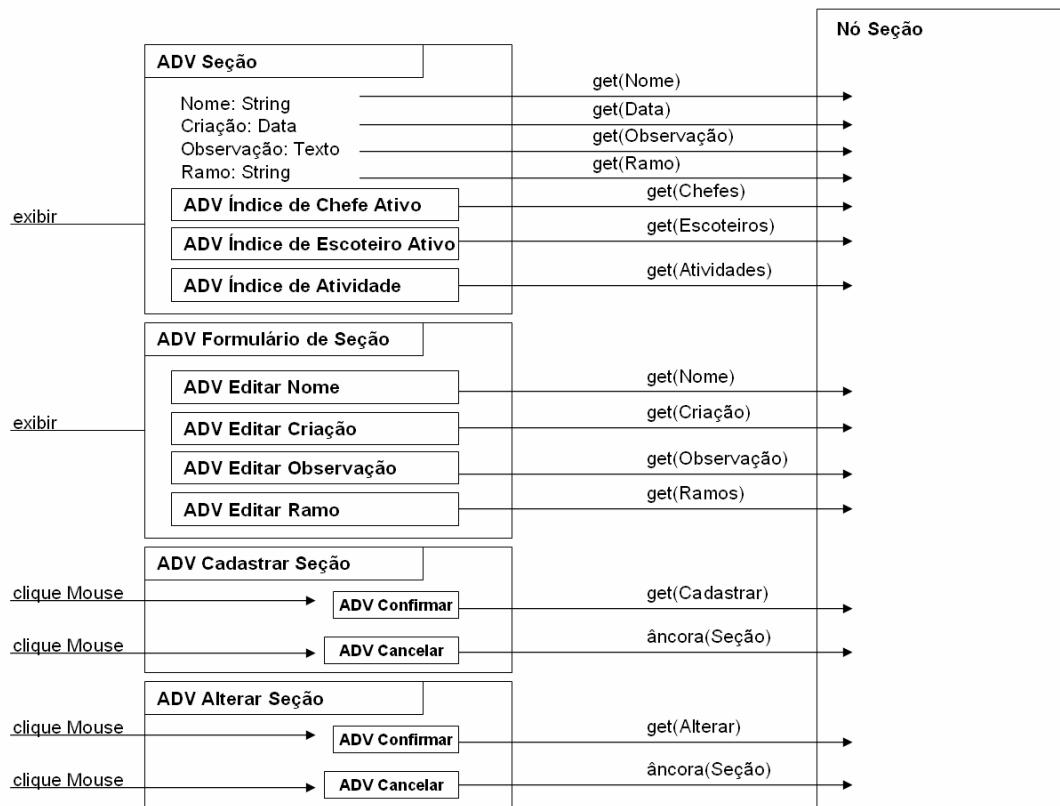


Figura 74: Diagrama de configuração do ADO Seção.

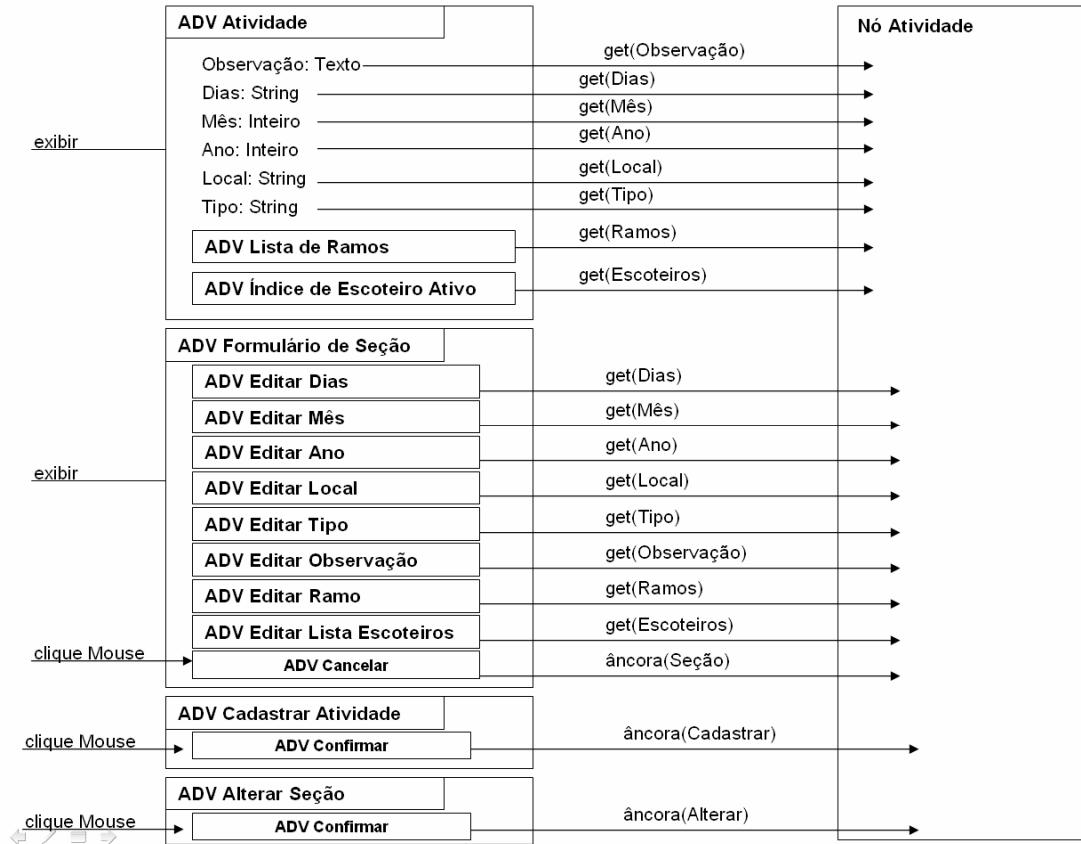


Figura 75: Diagrama de configuração do ADO Atividades.



Figura 76: Diagrama de configuração do ADO Formulário de Atividades.

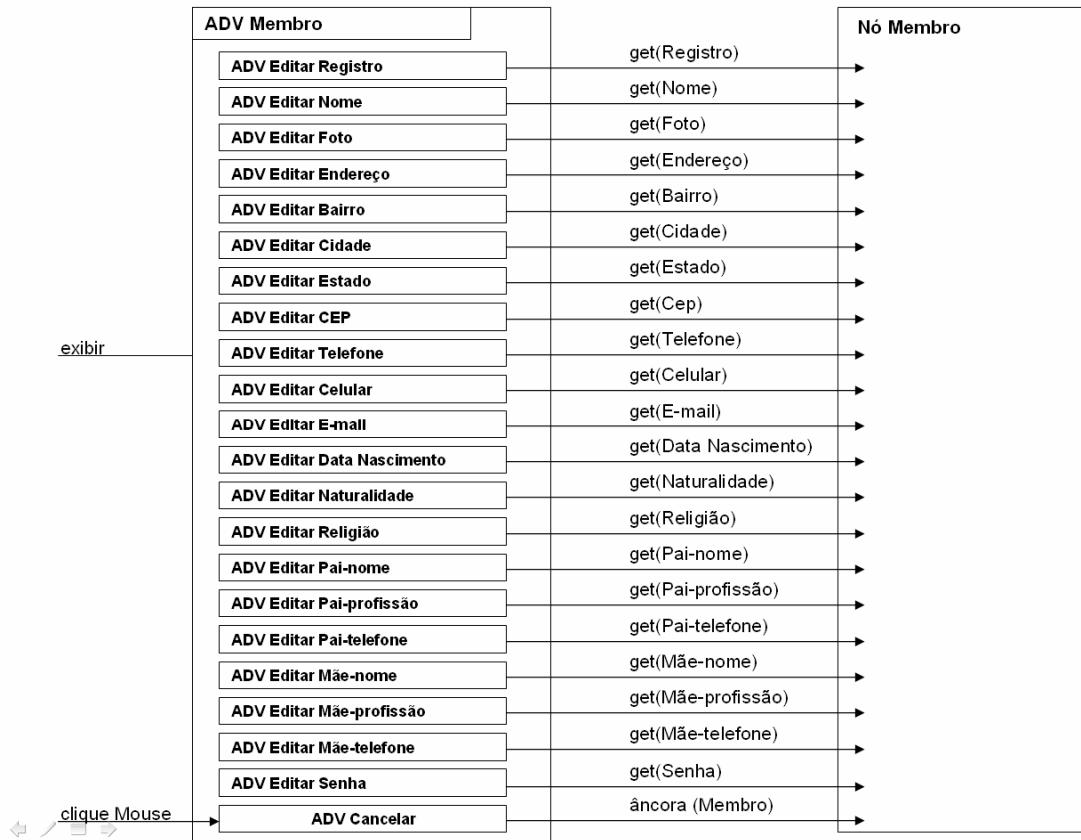


Figura 77: Diagrama de configuração do ADO Formulário Membro.

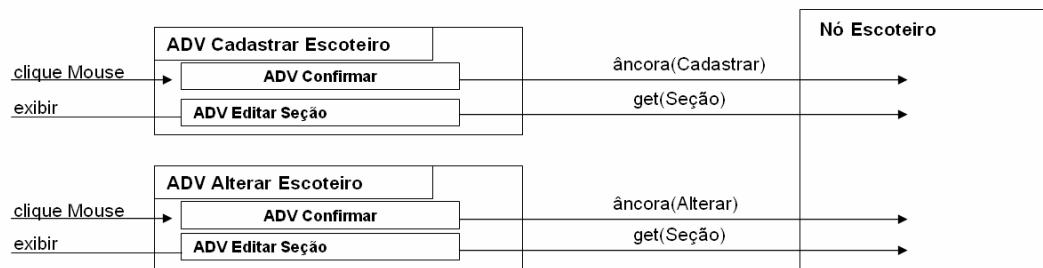


Figura 78: Diagrama de configuração do ADO Escoteiro.

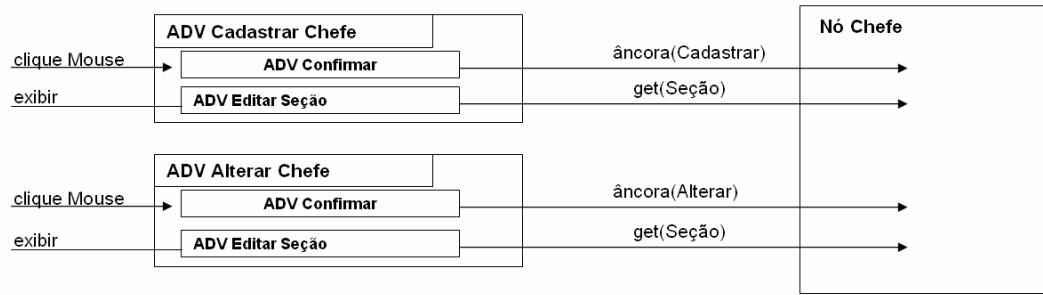


Figura 79: Diagrama de configuração do ADO Chefe.

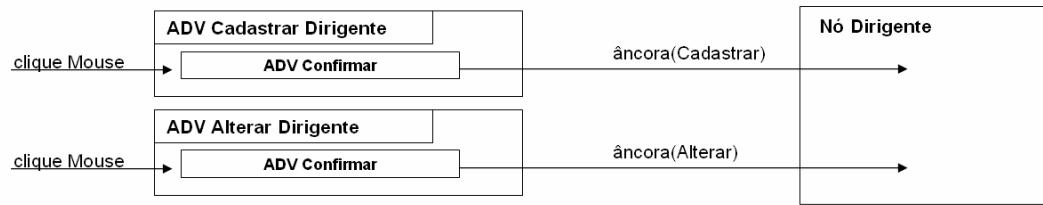


Figura 80: Diagrama de configuração do ADO Dirigente.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.