

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**  
**CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**  
**DISSERTAÇÃO DE MESTRADO**

**WEB-SEMP: MÉTODO DE ELICITAÇÃO,  
MODELAGEM E PLANEJAMENTO PARA  
APLICAÇÕES WEB**

**DÊNIS LEONARDO ZANIRO**

**Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Sandra Camargo P. F. Fabbri**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciências da Computação, área de concentração: Engenharia de Software

São Carlos/SP  
**Maio/2008**

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

Z31wm

Zaniro, Dênis Leonardo.

Web-semp: método de elicitação, modelagem e planejamento para aplicações web / Dênis Leonardo Zaniro. -- São Carlos : UFSCar, 2010. 126 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2008.

1. Ciência da computação. 2. Engenharia de software. 3. Sistemas web. 4. Engenharia de requisitos. I. Título.

CDD: 004 (20ª)

**Universidade Federal de São Carlos**  
**Centro de Ciências Exatas e de Tecnologia**  
**Programa de Pós-Graduação em Ciência da Computação**

**“Web-SEMP: Método de Elicitação, Modelagem e  
Planejamento para Aplicações Web”**

**DENIS LEONARDO ZANIRO**

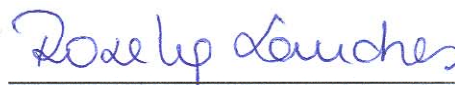
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

**Membros da Banca:**



---

**Profa. Dra. Sandra Camargo P. Ferraz Fabbri**  
**(Orientadora – DC/UFSCar)**



---

**Profa. Dra. Rosely Sanches**  
**(ICMC/USP)**



---

**Profa. Dra. Vera Maria Benjamim Werneck**  
**(IME/UERJ)**

**São Carlos**  
**Maio/2008**

*Aos meus pais, dedico este trabalho.*

# AGRADECIMENTOS

---

À *Prof<sup>a</sup>. Dr<sup>a</sup>. Sandra Fabbri*, pela excelente e constante orientação, pela confiança depositada em mim, paciência, atenção e apoio nos momentos mais difíceis.

Aos meus *pais*, por todo carinho e amor que sempre me deram.

A todos os meus *amigos de Araraquara*, pelos ótimos momentos juntos.

Aos meus *amigos do Laboratório de Pesquisa em Engenharia de Software (LaPES)*, Arnaldo, Daniel, Elis, Luiz e Renan pelo companheirismo.

A *Capes*, pelo apoio financeiro.

# SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>III</b>
<b>LISTA DE TABELAS.....</b>	<b>V</b>
<b>LISTA DE ABREVIATURAS.....</b>	<b>VI</b>
<b>RESUMO .....</b>	<b>VII</b>
<b>ABSTRACT .....</b>	<b>VIII</b>
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1. CONTEXTO .....	1
1.2. MOTIVAÇÃO E OBJETIVO .....	2
1.3. METODOLOGIA DE DESENVOLVIMENTO.....	3
1.4. ORGANIZAÇÃO DO TRABALHO .....	5
<b>2. ENGENHARIA DA WEB.....</b>	<b>6</b>
2.1. CONSIDERAÇÕES INICIAIS .....	6
2.2. CATEGORIAS E CARACTERÍSTICAS DE APLICAÇÕES WEB .....	6
2.2.1. <i>Características Relacionadas ao Produto.....</i>	<i>9</i>
2.2.2. <i>Características Relacionadas aos Usuários.....</i>	<i>10</i>
2.2.3. <i>Características Relacionadas à Equipe de Desenvolvimento .....</i>	<i>11</i>
2.3. PROCESSO DE DESENVOLVIMENTO .....	12
2.4. ANÁLISE DE REQUISITOS.....	14
2.4.1. <i>Métodos de Análise de Requisitos .....</i>	<i>15</i>
2.4.2. <i>O Modelo AWARE.....</i>	<i>17</i>
2.5. MODELAGEM DE REQUISITOS .....	22
2.5.1. <i>Métodos de Modelagem .....</i>	<i>24</i>
2.5.2. <i>WebML.....</i>	<i>26</i>
2.5.3. <i>Modelo de Casos de Uso Funcionais e Navegacionais.....</i>	<i>34</i>
2.6. PLANEJAMENTO .....	36
2.6.1. <i>Métodos para Planejamento de Hipermídia .....</i>	<i>36</i>
2.6.2. <i>Métodos para Planejamento Funcional.....</i>	<i>38</i>
2.7. CONSIDERAÇÕES FINAIS .....	40
<b>3. DEFINIÇÃO DA MÉTRICA WHP .....</b>	<b>41</b>
3.1. CONSIDERAÇÕES INICIAIS .....	41
3.2. EXTENSÃO DE ELEMENTOS DA WEBML .....	41
3.3. O PROCESSO DE DEFINIÇÃO DA MÉTRICA WHP .....	47
3.4. AVALIAÇÃO DA MÉTRICA WHP .....	53
3.5. CONSIDERAÇÕES FINAIS .....	58
<b>4. O MÉTODO WEB-SEMP .....</b>	<b>60</b>
4.1. CONSIDERAÇÕES INICIAIS .....	60
4.2. UMA VISÃO GERAL DO MÉTODO WEB-SEMP.....	60
4.3. FASE I – ELICITAÇÃO DE REQUISITOS E PLANEJAMENTO INICIAL .....	62
4.3.1. <i>Etapa 1: Elicitação de Requisitos .....</i>	<i>63</i>
4.3.2. <i>Etapa 2: Planejamento Inicial .....</i>	<i>65</i>
4.4. FASE II – APLICAÇÃO DA ESTRATÉGIA DE MAPEAMENTO .....	66
4.4.1. <i>Etapa 1: Identificação de Aspectos de Composição.....</i>	<i>67</i>
4.4.2. <i>Etapa 2: Identificação de Aspectos de Navegação.....</i>	<i>73</i>
4.4.3. <i>Etapa 3: Identificação de Aspectos de Funcionalidade .....</i>	<i>76</i>
4.5. FASE III – APLICAÇÃO DO MODELO DE MÉTRICA .....	77
4.5.1. <i>Etapa 1: Contagem de Caso de Uso Navegacional Estático .....</i>	<i>77</i>
4.5.2. <i>Etapa 2: Contagem de Caso de Uso Navegacional Dinâmico.....</i>	<i>78</i>

4.5.3. <i>Etapa 3: Contagem de Caso de Uso Funcional</i> .....	78
4.5.4. <i>Etapa 4: Avaliação de Complexidade Total da Aplicação Web</i> .....	79
4.6. CONSIDERAÇÕES FINAIS .....	79
<b>5. EXEMPLO DE APLICAÇÃO NO SISTEMA REAL DA MIREX .....</b>	<b>80</b>
5.1. CONSIDERAÇÕES INICIAIS .....	80
5.2. CARACTERIZAÇÃO DO ESTUDO DE CASO .....	80
5.3. APLICAÇÃO DA FASE I .....	81
5.3.1. <i>Etapa 1: Elicitação de Requisitos</i> .....	81
5.3.2. <i>Etapa 2: Planejamento Inicial</i> .....	90
5.4. APLICAÇÃO DA FASE II.....	90
5.4.1. <i>Etapa 1: Identificação de Aspectos de Composição</i> .....	90
5.4.2. <i>Etapa 2: Identificação de Aspectos de Navegação</i> .....	101
5.4.3. <i>Etapa 3: Identificação de Aspectos de Funcionalidade</i> .....	103
5.5. APLICAÇÃO DA FASE III.....	105
5.5.1. <i>Etapa 1: Contagem de Caso de Uso Navegacional Estático</i> .....	105
5.5.2. <i>Etapa 2: Contagem de Caso de Uso Navegacional Dinâmico</i> .....	105
5.5.3. <i>Etapa 3: Contagem de Caso de Uso Funcional</i> .....	106
5.5.4. <i>Etapa 4: Avaliação de Complexidade Total da Aplicação Web</i> .....	106
5.6. CONSIDERAÇÕES FINAIS .....	106
<b>6. CONCLUSÕES.....</b>	<b>108</b>
6.1. CONTRIBUIÇÕES DO TRABALHO.....	110
6.2. TRABALHOS FUTUROS .....	111
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>112</b>
<b>APÊNDICE A .....</b>	<b>117</b>

# LISTA DE FIGURAS

Figura 1: Representação da motivação do trabalho desenvolvido. ....	3
Figura 2: Metodologia usada durante o trabalho. ....	4
Figura 3: Modelo de Processo Genérico para aplicações Web. [Perrone et al., 2005b].	12
Figura 4: Meta-Modelo AWARE para aplicações Web. [Bolchini & Paolini, 2004]....	22
Figura 5: Domínios de Modelagem para Aplicações Web. [Adaptado de Kappel et al., 2006].....	23
Figura 6: Exemplo de Modelo Estrutural e Código XML.....	27
Figura 7: Notação Gráfica e Código XML para uma <i>Data Unit</i> . ....	28
Figura 8: Notação Gráfica e Código XML para uma <i>MultiData Unit</i> . ....	28
Figura 9: Notação Gráfica e Código XML para uma <i>Index Unit</i> . ....	29
Figura 10: Notação Gráfica e Código XML para uma <i>Scroller Unit</i> . ....	29
Figura 11: Notação Gráfica e Código XML para uma <i>Entry Unit</i> . ....	30
Figura 12: Notação Gráfica e Código XML para uma Página. ....	30
Figura 13: Notação Gráfica e Código XML para Sub-Páginas AND. ....	31
Figura 14: Notação Gráfica e Código XML para Sub-Páginas OR. ....	32
Figura 15: Notação Gráfica e Código XML para <i>Links</i> Contextuais. ....	33
Figura 16: Notação Gráfica e Código XML para <i>Links</i> Não-Contextuais. ....	33
Figura 17: Notação Gráfica para uma <i>Image Unit</i> e Código XML Correspondente.....	43
Figura 18: Notação Gráfica para uma <i>MultiImage Unit</i> e Código XML Correspondente. ....	44
Figura 19: Notação Gráfica para uma <i>Animation Unit</i> e Código XML Correspondente. ....	45
Figura 20: Notação Gráfica para uma <i>Text Unit</i> e Código XML Correspondente.....	45
Figura 21: Trecho do Modelo de Hipertexto para a aplicação Web Tapetes-SC.....	46
Figura 22: O método Web-SEMP no contexto do Processo de Desenvolvimento Web.	61
Figura 23: Diagrama GTR para o Gerente de um Fornecedor de Tapetes. ....	65
Figura 24: Diagrama GTR Numerado para o Fornecedor de Tapetes.....	67
Figura 25: Exemplo de Mapeamento de Requisitos de Conteúdo na WebML Estendida. ....	70
Figura 26: Exemplo de Mapeamento de Requisitos de Apresentação na WebML Estendida. ....	71
Figura 27: Exemplo de Mapeamento de Requisitos de Interação na WebML Estendida. ....	72
Figura 28: Exemplo de Aplicação da Etapa 2 para o diagrama GTR do Fornecedor de Tapetes.....	75
Figura 29: Diagrama GTR para o Diretor da Mirex.....	84
Figura 30: Diagrama GTR para o Representante de <i>Marketing</i> . ....	85
Figura 31: Diagrama GTR para o Cliente Mirex.....	86
Figura 32: Diagrama GTR para o Diretor da Mirex (Visão de Balconistas).....	87
Figura 33: Diagrama GTR para o Representante de Balconista.....	87
Figura 34: Diagrama GTR para o Diretor da Mirex (Visão de Gerenciamento).....	89
Figura 35: Diagrama GTR Numerado para o Cliente Mirex.....	91
Figura 36: Diagrama GTR Numerado para o Representante de <i>Marketing</i> . ....	91
Figura 37: Diagrama GTR Numerado para o Diretor da Mirex. ....	92
Figura 38: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Empresa”.94	
Figura 39: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Produto”. 96	



---

Figura 40: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Tecnologia”.  
..... 97

Figura 41: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Formiga”. 98

Figura 42: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Métodos”.99

Figura 43: Sub-modelo de Composição após a Aplicação da Etapa 1 para “Notícias”.  
..... 100

Figura 44: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Contato”.  
..... 100

Figura 45: Modelo de Hipertexto após a aplicação dos Passos 1 e 2. .... 102

Figura 46: Modelo de Casos de Uso Navegacionais para a Mirex..... 103

Figura 47: Modelo de Casos de Uso Funcionais e Navegacionais para a Visão Pública da Mirex..... 104

# LISTA DE TABELAS

---

---

Tabela 1: Níveis de Complexidade para uma Página em Web Points. [Cleary, 2000] ..	37
Tabela 2: Pontos Web para cada Nível de Complexidade. [Cleary, 2000] .....	37
Tabela 3: Pontos de Complexidade para Imagens.....	49
Tabela 4: Pontos de Complexidade para Vetorização de Imagens.....	49
Tabela 5: Pontos de Complexidade para Animações. ....	50
Tabela 6: Pontos de Complexidade para Textos. ....	51
Tabela 7: Pontos de Complexidade para Entradas e Saídas de Dados. ....	52
Tabela 8: Pontos de Complexidade para Relação Estrutural.....	52
Tabela 9: Dados de Aplicações Web usadas na Avaliação. ....	54
Tabela 10: Pontos de Complexidade Atribuídos a cada Aplicação Web. ....	54
Tabela 11: Cálculo de Esforço e MRE para cada Projeto Web.....	56
Tabela 12: Cálculo de Esforço em Web Points e Valor de MRE para Sites Web Estáticos.....	57
Tabela 13: Modelos de Estimativa Gerados por Mendes et al. (2001).....	58
Tabela 14: Exemplo de Atividades Não-Técnicas Previstas.....	66
Tabela 15: Exemplo de um Formulário TRS.....	68
Tabela 16: Visões da Aplicação e <i>Stakeholders</i> Correspondentes. ....	82
Tabela 17: Formulário TRS Preenchido para a Visão Pública da Mirex. ....	93

# LISTA DE ABREVIATURAS

---

---

Web-SEMP	Analysis of Web Applications REquirements
WHP	Web Hypermedia Points
AWARE	Analysis of Web Applications REquirements
MODFM	Mockup-driven Fast-prototyping Methodology for Web Requirements Engineering
GBRAM	Goal-Based Requirements Analysis Method
DSL	Definitional Specification Language
WebML	Web Modeling Language
HDM	Hypermedia Design Method
OOHDM	Object-Oriented Hypermedia Design Method
WAE	Web Application Extension
PCU	Pontos por Casos de Uso

# RESUMO

---

---

**Cenário:** *Aplicações Web diferem significativamente de sistemas de software convencionais, impactando a Engenharia de Requisitos (ER) para a Web. A maioria dos métodos propostos na literatura para atender esses sistemas é utilizada de forma isolada e não integra algumas das práticas e princípios da Engenharia de Software. Além disso, não se encontram diretrizes que auxiliem na construção de modelos conceituais de projeto mais próximos de especificações de requisitos e, ao mesmo tempo, expressivos, de forma a permitir a aplicação de métricas de software na fase de planejamento.* **Objetivo:** *Com base nesse contexto, o objetivo deste trabalho foi definir o método Web-SEMP – Web System Elicitation, Modeling and Planning – para auxiliar nas atividades de elicitação, modelagem e planejamento para aplicações Web. O método determina uma abordagem para criar um modelo conceitual com base nos requisitos, bem como define uma métrica, denominada WHP – Web Hypermedia Points – para avaliar o tamanho de aplicações Web.* **Método:** *As técnicas de análise de requisitos e de planejamento mais citadas na literatura foram analisadas com o objetivo de utilizá-las e estendê-las para compor o método aqui apresentado. Da literatura foram adotadas as técnicas AWARE e WebML para elicitação e modelagem, e a métrica Pontos por Casos de Uso para a atividade de planejamento.* **Resultados:** *O método Web-SEMP foi aplicado no desenvolvimento de uma aplicação Web real e os resultados fornecem evidências de que ele contribui efetivamente para sistematizar o processo de ER para a Web. Para avaliar a efetividade da métrica WHP, ela foi aplicada em alguns sites Web e os resultados obtidos refletiram, em termos proporcionais, as horas gastas para desenvolvê-los.* **Conclusões:** *De acordo com os resultados, é possível concluir que o método Web-SEMP contribui com a melhoria da qualidade da ER para Web, e a métrica WHP representa um primeiro passo para a obtenção de modelos de estimativa de esforço e de custo mais precisos.*

# ABSTRACT

---

---

**Background:** *Web applications differ from traditional applications in several significant ways, affecting the Requirements Engineering for Web (WebRE). The most WebRE methods proposed in the scientific literature are used in isolation and don't merge some of all practices and principles of Software Engineering. Besides, there aren't guidelines to tie up requirements specifications with conceptual design models and, at the same time, to support the construction of expressive models in such a way that they allow the metric application in the planning phase.* **Goal:** *based on this context, the goal of this work was to define a method, named Web-SEMP – Web System Elicitation, Modeling and Planning – proposed to help the elicitation, modeling and planning activities for Web applications. The proposed method defines both an approach to elaborate a conceptual model based on requirements, and a metric, named WHP – Web Hypermedia Points – to evaluate the size of Web applications.* **Method:** *the most quoted methods in the literature were analyzed aiming to reuse and to extend them to compose the method here proposed. From literature, the proposed method adopts both the AWARE and WebML techniques for elicitation and modeling activities, respectively, and the Use Case Points method for planning activity.* **Results:** *The Web-SEMP method was applied in the real development of a Web application and the results give insights of its contribution to conduct the WebRE process in a more systematic way. In order to evaluate the effectiveness of the WHP metric, it was applied in some implemented Web sites and the obtained results reflected, in proportional terms, the hours spent to develop them.* **Conclusions:** *According to the results, it could be observed that the Web-SEMP method enables to improve the quality of WebRE, and the WHP metric represents a first step towards more precise effort and cost estimation models.*

# CAPÍTULO 1

## INTRODUÇÃO

---

### **1.1. Contexto**

Uma das tarefas mais difíceis enfrentadas por um engenheiro de software é entender as reais necessidades de clientes e usuários, traduzindo este entendimento num conjunto de especificações de requisitos. Nesse contexto, a Fase de Engenharia de Requisitos (ER) se torna fundamental, uma vez que as maiores dificuldades relacionadas ao desenvolvimento devem ser tratadas nessa etapa [Zhang et al., 2003].

Alguns problemas enfrentados pela ER são originados a partir da falta de controle sobre aquilo que precisa ser construído. Dentre esses, podem-se citar: pouco esforço despendido no levantamento de informações junto ao usuário, documentação pobre sobre os requisitos obtidos, requisitos altamente voláteis, especificações ambíguas ou incorretas de requisitos e tendência a iniciar rapidamente o processo de desenvolvimento do software [Kappel et al., 2006]. Outros problemas, relacionados aos anteriores, são decorrentes de falhas de comunicação entre *stakeholders* e projetistas e dificuldades de compreensão, pois, na grande maioria das vezes, clientes e usuários não possuem um entendimento preciso e detalhado de suas necessidades.

Embora os problemas citados correspondam ao processo de ER de forma geral, no contexto de aplicações Web, tais dificuldades são encontradas com maior frequência, devido às restrições impostas pelas características específicas do ambiente Web [Deshpande et al., 2002]. Assim, diversos métodos são propostos na literatura para dar suporte às atividades de ER para a Web, entretanto a maioria deles é aplicada de maneira isolada e não integra algumas das práticas e princípios fundamentais da engenharia de software [Perrone et al., 2005b]. Além disso, na literatura, não se encontram diretrizes para conduzir a equivalência entre um modelo inicial de requisitos e modelos conceituais de projeto. A ausência de métodos que estabeleçam claramente esta correspondência tem levado a limitações na utilidade desses trabalhos na prática.

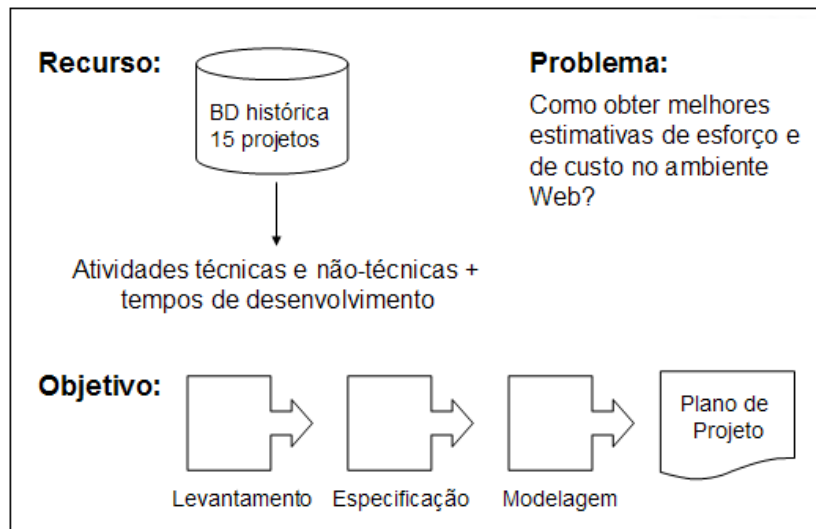
Métodos de ER para a Web devem levar em consideração que muitos analistas Web e *stakeholders* do projeto têm pouco ou nenhum conhecimento técnico, portanto tais métodos não podem exigir muito esforço de treinamento para que sejam adotados e efetivamente integrados em práticas atuais [Zhang et al., 2003] [Pressman, 2006]. Os modelos e notações propostas pelo método devem ser leves e intuitivas para que sirvam como um canal de comunicação entre todos os *stakeholders* envolvidos, mas, ao mesmo tempo, suas representações devem ser completas de forma a permitir a aplicação de métricas de software durante a atividade subsequente de planejamento.

A atividade de planejamento representa um dos passos cruciais no início de qualquer projeto de software [Pfleeger, 2004]. Estimativas de custo e de esforço imprecisas podem ter um impacto negativo significativo na reputação e competitividade de uma companhia, influenciando diretamente no sucesso do projeto de desenvolvimento Web como um todo. Dessa forma, é fundamental que as atividades de especificação e modelagem de requisitos propiciem a avaliação da complexidade de desenvolvimento, com base nos requisitos levantados. É importante observar que técnicas de planejamento devem levar em consideração tanto características de hipermídia quanto aspectos de funcionalidade que compõem uma aplicação Web.

## **1.2. Motivação e Objetivo**

Considerando-se o contexto apresentado anteriormente e motivado pelos dados históricos acerca de 15 aplicações Web concluídas em uma pequena empresa de desenvolvimento de software, o objetivo deste trabalho foi definir um método que apoiasse as atividades de elicitação de requisitos, modelagem e planejamento para aplicações Web.

Apesar de a empresa registrar os tempos de todas as atividades conduzidas – técnicas e não-técnicas – durante o desenvolvimento de uma aplicação Web, não se conhecia uma forma de se chegar a representações conceituais adequadas a fim de possibilitar obter melhores estimativas de esforço e de custo no ambiente Web, conforme representado na Figura 1. É importante ressaltar que essa necessidade fazia parte de um plano de ações dessa empresa, que buscava implementar melhorias em seu processo e chegar a estimativas de prazo e de custo mais realísticas.



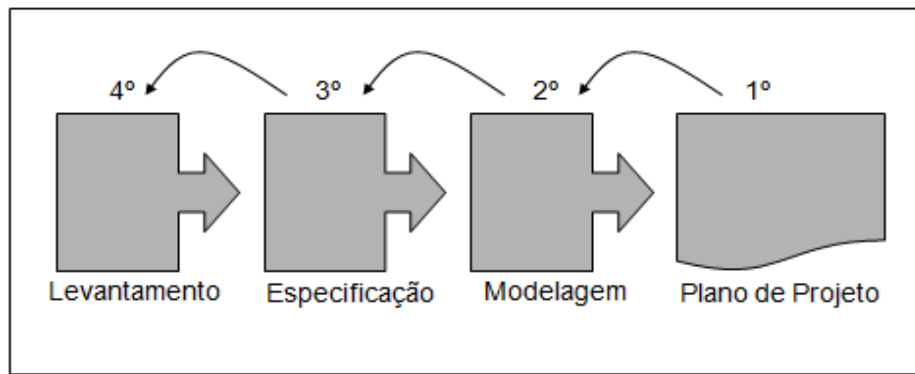
**Figura 1: Representação da motivação do trabalho desenvolvido.**

Assim, o objetivo principal do trabalho foi resolver o problema mostrado na Figura 1 determinando um caminho que abrangesse as etapas de levantamento, especificação e modelagem de requisitos e permitisse, como resultado, a construção de um plano de projeto mais realístico. Além disso, Para guiar o analista a efetuar a transição entre as etapas, definiu-se também um conjunto de diretrizes levando à construção de modelos que dessem apoio efetivo a estimativas de prazo e de custo melhores.

### **1.3. Metodologia de Desenvolvimento**

De acordo com a motivação apresentada anteriormente, a elaboração desta proposta seguiu o caminho inverso daquele representado na Figura 1. Inicialmente, cada etapa do processo de ER para a Web foi tratada como uma caixa-preta, e a pesquisa foi conduzida de “trás para frente”, conforme representado na Figura 2. Uma vez que definiram-se as atividades que deveriam constituir a última etapa – etapa de planejamento –, foi possível especificar atividades em etapas anteriores para atender exatamente às necessidades da etapa subsequente. Em essência, uma vez que as saídas de uma determinada etapa foram conhecidas, definiu-se um conjunto de atividades que fornecesse as entradas necessárias para gerar tal saída. Assim, essa abordagem permitiu uma avaliação mais completa das necessidades de cada etapa, com base somente nos tipos de elementos de entrada exigidos pela etapa subsequente.





**Figura 2: Metodologia usada durante o trabalho.**

Para elaborar o planejamento, primeiramente os dados históricos que a companhia possuía foram analisados e, a partir dessa análise, foram levantados todos os elementos que influenciam na complexidade de desenvolvimento de uma aplicação Web. Assim, selecionou-se um modelo que pudesse representar tais elementos de uma aplicação Web de uma maneira simples e, ao mesmo tempo, abrangente, para que, a partir do qual fosse possível avaliar o esforço de implementação desses elementos. Para construir esses modelos, adotou-se a linguagem WebML como método de modelagem; no entanto, observando-se que alguns elementos de conteúdo essenciais de uma aplicação Web ainda não eram representados pela WebML, estendeu-se esse método a partir da elaboração de algumas notações novas.

Por meio da WebML estendida, definida neste trabalho, todos os projetos cujos dados eram possuídos pela empresa foram modelados. A partir do estudo dos resultados obtidos, definiu-se a métrica WHP, para dar suporte a estimativas de esforço mais realísticas, considerando-se apenas a complexidade de desenvolvimento de elementos específicos de Web. Além disso, observando-se a necessidade de se avaliar também a complexidade funcional de uma aplicação, adotou-se a métrica PCU – Pontos por Casos de Uso –, para avaliar a complexidade de cada elemento funcional de uma aplicação Web.

Uma vez que os componentes necessários foram definidos para dar suporte à aplicação do planejamento, decidiu-se identificar um modelo que pudesse auxiliar na construção de tais componentes. Para essa finalidade, adotou-se o modelo AWARE, a partir do qual pode-se usar diagramas GTR – *Goal, Task, Requirements* –, que descrevem objetivos, tarefas e requisitos de todos os *stakeholders*, conforme é definido

pelo modelo. Como esse modelo define uma abordagem orientada a objetivos, adotou-se também o modelo de casos de uso funcionais e navegacionais, uma vez que tal modelo é considerado uma abordagem orientada a tarefas, permitindo complementar a análise orientada a objetivos.

Finalmente, para guiar a transição entre requisitos e modelos de análise, foi definido um conjunto de diretrizes para dar apoio à construção de modelos WebML e casos de uso funcionais e navegacionais, considerando-se os objetivos e requisitos descritos nos diagramas GTR. Observa-se, portanto, que o conjunto de diretrizes foi definido com base nos elementos Web que deveriam constar nos modelos de análise e que já eram conhecidos desde o início da pesquisa. Portanto, essa abordagem permitiu derivar apenas as atividades que estavam de acordo com os elementos Web necessários para a aplicação do planejamento.

#### **1.4. Organização do trabalho**

O presente trabalho está organizado em cinco capítulos, além da seção referente às Referências Bibliográficas e do Apêndice A.

Embora a pesquisa tenha sido conduzida de maneira inversa, conforme foi explicado na Seção 1.3, partindo do planejamento e chegando ao levantamento de requisitos, a apresentação dos capítulos não corresponde a essa mesma ordem, sendo que eles apresentam o embasamento teórico necessário para que o leitor compreenda a proposta toda, a qual é apresentada no Capítulo 4.

Os capítulos estão organizados da seguinte forma: no Capítulo 2, apresentam-se os principais conceitos e trabalhos relacionados às atividades de análise de requisitos, modelagem e planejamento para a Web; no Capítulo 3, apresenta-se a métrica proposta, denominada WHP (*Web Hypermedia Points*); no Capítulo 4, apresenta-se o método Web-SEMP – *Web System Elicitation, Modeling and Planning*; no Capítulo 5, descreve-se um estudo de caso relacionado à utilização do método proposto; no Capítulo 6, são apresentadas as conclusões e, finalmente, no Apêndice A, são apresentadas as tabelas de pontos preenchidas com os valores resultantes da aplicação da métrica WHP em alguns sites Web.

# CAPÍTULO 2

## ENGENHARIA DA WEB

---

### ***2.1. Considerações Iniciais***

Neste capítulo são discutidos os principais conceitos e trabalhos relacionados às três etapas do processo de construção de uma aplicação Web que são apoiadas pelo método Web-SEMP. São elas: Análise de Requisitos, Modelagem e Planejamento. Essas três etapas estão inseridas no contexto do processo genérico de Engenharia da Web.

A Engenharia da Web pode ser definida como uma abordagem sistemática e disciplinada para criar aplicações Web de alta qualidade [Pressman, 2006]. Combinando estratégias, práticas e métodos especializados, essa disciplina lida com todos os aspectos de desenvolvimento de aplicações Web, desde as fases de concepção e desenvolvimento até as fases de codificação, testes e manutenção contínua. Dessa forma, a Engenharia da Web pode ser considerada como uma sub-área independente da Engenharia de Software, responsável pelo estudo de abordagens sistemáticas e quantificáveis aplicadas ao desenvolvimento, operação e manutenção de aplicações Web [Kappel et al., 2006].

O restante deste capítulo está organizado da seguinte forma: na Seção 2.2 são apresentadas as categorias e algumas das principais características de aplicações Web. Na Seção 2.3 é discutido um modelo de processo genérico para aplicações Web. Nas Seções 2.4, 2.5 e 2.6, são apresentadas, respectivamente, as etapas de Análise de Requisitos, Modelagem e Planejamento para aplicações Web. Finalmente, na Seção 2.7 as considerações finais são apresentadas.

### ***2.2. Categorias e Características de Aplicações Web***

Aplicações Web formam, normalmente, uma complexa matriz de conteúdo e funcionalidade para um conjunto diversificado de usuários finais. De acordo com [Pressman, 2006], aplicações Web envolvem uma combinação de publicação impressa e desenvolvimento de software, comercialização e computação, comunicações internas e relações externas, e arte e tecnologia.

Devido a diversas propriedades específicas, há pouca discussão sobre o fato de que aplicações Web são diferentes de muitas outras categorias de software de computador [Pressman, 2006]. Uma aplicação Web reside em uma rede e precisa servir às necessidades de uma comunidade diversificada de clientes, o que significa que um grande número de usuários pode ter acesso à aplicação ao mesmo tempo. A função principal de muitas aplicações Web é usar os recursos de hipermídia para apresentar diversos tipos de conteúdo ao usuário final, incluindo texto, imagens, gráficos, áudio e vídeo [Pressman, 2006]. Além disso, em muitos casos, tais aplicações são usadas para dar acesso a informações que existem em bancos de dados que não faziam parte originariamente de um ambiente baseado na Web.

A evolução contínua é outra característica que permite distinguir uma aplicação Web de uma aplicação de software convencional [Pressman, 2006]. Enquanto que esta última evolui, normalmente, por meio de uma série de versões cuidadosamente planejadas, as aplicações Web podem evoluir continuamente. A necessidade de atualizações constantes de conteúdo, mudanças em aspectos da interface em nível de apresentação e a necessidade de migrar para novas tecnologias que são desenvolvidas rapidamente exemplificam a natureza do processo de manutenção contínuo requerido por uma aplicação Web.

A alta competitividade no mercado exige que ciclos de vida de desenvolvimento no ambiente Web sejam cada vez menores, tornando difícil, na grande maioria das vezes, a aplicação de abordagens seqüenciais e rigorosamente disciplinadas de desenvolvimento. As aplicações Web freqüentemente exigem um prazo para serem colocadas no mercado que pode variar de dias ou semanas [Reifer, 2002] [Mendes et al., 2005a]. Outras características ainda incluem aspectos de qualidade tais como segurança, alta disponibilidade, padrões de navegação consistentes, previsibilidade de informação, estética, escalabilidade, desempenho e qualidade do conteúdo, no que diz respeito ao seu nível de atualização, consistência e confiabilidade.

Essas propriedades gerais se aplicam a todas as aplicações Web, entretanto com diferentes graus de influência, o que sugere uma categorização de tais aplicações. Embora haja diversas taxonomias propostas na literatura para classificar as aplicações Web em si, podem ser identificadas duas amplas categorias, de acordo com o seu estilo de interação [Deshpande & Hansen, 2001] [Bolchini & Paolini, 2004]:

- *Aplicações de Hipermissão:* são as aplicações de hipertexto e multimídia, formadas basicamente por um conjunto de páginas e *links*. Cada página é constituída por elementos de conteúdo, os quais são apresentados aos usuários, em resposta a alguma solicitação. Nessa categoria, os usuários basicamente navegam pelas páginas, exploram diversas categorias de conteúdo por meio de botões ou menus e selecionam *links* que dão acesso a outras páginas ou aplicações. Em aplicações Web de hipermissão, o conteúdo oferecido e a arquitetura navegacional são os elementos-chave de modelagem.
- *Aplicações Transacionais:* são as aplicações que têm por objetivo prover maior interatividade, permitindo ao usuário ativar operações, enviar e atualizar dados, receber notificações e mensagens de sistema e efetuar transações, da mesma forma que nos sistemas de informação convencionais. Essas aplicações normalmente estão associadas a sistemas de bancos de dados que permitem a operação e a manutenção eficientes de uma grande quantidade de conteúdo, por meio da realização de consultas estruturadas.

Portanto, o paradigma transacional enfatiza os serviços e operações providas pela aplicação Web enquanto que o paradigma de hipermissão foca no conteúdo e aspectos navegacionais e de apresentação disponíveis aos usuários. Nos dias de hoje, as aplicações Web complexas, no entanto, combinam esses dois paradigmas de interação [Bolchini & Paolini, 2004]. Mesmo sites Web pequenos, constituídos de pouco mais do que um conjunto de arquivos de hipertexto ligados que apresentam informação, precisam ser gerenciados ou, ainda, integrados com bancos de dados corporativos e outras aplicações de negócio. Sites Web de *e-commerce*, *e-business* e *e-banking* são alguns exemplos típicos dessa combinação de estilos de interação.

Nas próximas sub-seções serão apresentadas as principais características que devem ser consideradas durante o desenvolvimento de uma aplicação Web. Essas características estão relacionadas ao produto de software final, aos usuários e à equipe de desenvolvimento.

### 2.2.1. Características Relacionadas ao Produto

As características relacionadas ao produto constituem os principais blocos de construção de uma aplicação Web, consistindo de *conteúdo*, *estrutura de hipertexto* (estrutura navegacional) e *apresentação* (interface do usuário) [Kappel et al., 2006]. De acordo com requisitos de usuários, o conteúdo informacional provido por uma aplicação Web pode ser representado por meio de tabelas, textos, representações gráficas, animações, áudio ou vídeo. Dependendo do domínio de aplicação, o conteúdo não é apenas submetido a atualizações freqüentes, mas também à aplicação de métricas que avaliem seu nível de exatidão, consistência, atualização e confiabilidade. Esses fatores devem ser levados em consideração durante todo o processo de desenvolvimento de uma aplicação Web.

Assim, a qualidade do conteúdo é um fator crítico para a aceitação de uma aplicação Web no mercado, representando um grande desafio aos desenvolvedores e web *designers*, os quais devem ser capazes de garantir a qualidade dos dados apesar do grande volume e da alta freqüência de atualizações, como acontece comumente no ambiente Web.

A estrutura de hipertexto de uma aplicação Web forma a base para a organização e apresentação de informações [Pressman, 2006]. Elementos básicos que constituem um modelo de hipertexto são nodos, *links* e âncoras. Um nodo pode ser visto com uma unidade de informação unicamente identificável, representada normalmente por uma página na aplicação Web. Um *link* é o caminho unidirecional de um nodo a outro, e uma âncora é a área dentro do conteúdo de um nodo que indica a origem ou o destino de um *link* (uma âncora pode ser representada por uma seqüência de palavras em um texto ou um objeto gráfico).

A característica essencial de um modelo de hipertexto é a sua natureza não-linear [Kappel et al, 2006]. Usuários podem navegar livremente pelo espaço de informação, de acordo com os seus interesses e conhecimento prévio, seguindo, dessa forma, os caminhos navegacionais que melhor correspondem às suas expectativas e objetivos. Essa natureza não-linear leva a dois problemas fundamentais que devem ser tratados durante o desenvolvimento de uma aplicação Web: desorientação e sobrecarga cognitiva. O primeiro problema se refere à dificuldade apresentada pelo usuário para se localizar em uma determinada porção da aplicação Web. Já o segundo problema diz

respeito à dificuldade que o usuário apresenta, muitas vezes, para memorizar e entender uma grande quantidade de informação disponibilizada de forma desorganizada pela aplicação Web.

No nível de apresentação, há duas características especiais que devem ser levadas em consideração: estética e facilidade de uso. Os aspectos estéticos são inegavelmente parte da atração de uma aplicação Web e, em muitos casos, podem determinar o sucesso ou o fracasso de uma aplicação Web, em particular para aplicações de *e-commerce* [Pressman, 2006]. No que diz respeito à facilidade de uso, é essencial que as aplicações Web sejam “auto-explicativas”, isto é, deve ser possível usar uma aplicação Web sem a necessidade de documentação.

### **2.2.2. Características Relacionadas aos Usuários**

Em comparação com as aplicações tradicionais, a utilização de aplicações Web é extremamente heterogênea, o que significa que diferentes grupos de usuários podem ter acesso a uma aplicação Web, possuindo diferenças significativas em termos de habilidades, conhecimento, preferências e interesses [Deshpande & Hansen, 2001]. Dessa forma, o número e a categoria de usuários não podem ser prognosticados de maneira confiável da mesma forma que nas aplicações e sistemas convencionais. Entretanto, um exemplo representativo do público-alvo deve ser definido para que a aplicação satisfaça todos os objetivos de usuários e clientes.

As aplicações Web normalmente oferecem serviços não apenas para um tipo específico de dispositivos, mas para dispositivos com muitas especificações diferentes (por exemplo, tamanhos de monitores diferentes, capacidade de memória, sistema operacional utilizado, etc.).

O número elevado de versões de navegadores diferentes é outro problema que precisa ser tratado pelos desenvolvedores, visto que cada versão apresenta suas próprias funcionalidades e restrições, ocasionando dificuldades para a elaboração de uma interface de usuário consistente. Além disso, vale ressaltar que usuários podem configurar navegadores individualmente, o que pode afetar diretamente no desempenho, nas propriedades de transação e nas possibilidades de interação oferecidas por uma aplicação Web.

### 2.2.3. Características Relacionadas à Equipe de Desenvolvimento

O desenvolvimento de uma aplicação Web bem-sucedida exige diversas habilidades, o que requer a adoção de uma abordagem multidisciplinar [McDonald & Welland, 2001]. Os principais papéis que devem ser distribuídos entre os membros da equipe de Engenharia da Web são [Pressman, 2006]:

- **Desenvolvedores/Provedores de Conteúdo:** como a grande maioria das aplicações Web é inerentemente guiada por conteúdo, um provedor de conteúdo tem a função de elaborar ou coletar todas as informações que constituirão o espectro de conteúdo a ser fornecido pela aplicação.
- **Editor da Web:** uma vez que o conteúdo a ser disponibilizado foi coletado pelo provedor de conteúdo, é necessário que algum membro da equipe – o editor da Web – organize todas as informações relevantes para inclusão na aplicação Web, agindo como elemento de ligação entre desenvolvedores e *web designers* (equipe técnica) e provedores de conteúdo.
- **Engenheiro da Web:** é responsável por uma grande variedade de atividades durante o desenvolvimento de uma aplicação Web, incluindo as atividades de elicitação e modelagem de requisitos, projeto arquitetural, navegacional e de interface. Engenheiros da web são representados, normalmente, por desenvolvedores e *web designers*, os quais devem ter um conhecimento sólido sobre tecnologias de implementação e arquiteturas cliente/servidor.
- **Especialistas no domínio do negócio:** geralmente são os membros da equipe responsáveis por estabelecer reuniões e manter contato com clientes e usuários. Um especialista no domínio do negócio deve ter um conhecimento detalhado de todas as questões relativas às metas, objetivos e requisitos de negócio referentes à aplicação Web a ser desenvolvida.

É importante observar que, na prática, cada um desses papéis é, muitas vezes, compartilhado por diversos membros da equipe de Engenharia da Web, ou ainda um único membro pode assumir papéis diferentes durante o desenvolvimento. Em equipes pequenas, é comum, por exemplo, que *web designers* e desenvolvedores desempenhem



também o papel de provedores de conteúdo, editores da Web ou ainda especialistas no domínio do negócio [McDonald & Welland, 2001].

Vale ressaltar também a faixa etária da maioria dos desenvolvedores e *web designers*, os quais são, em média, mais novos e, freqüentemente, menos experientes em relação aos desenvolvedores de software tradicional [Kappel et al., 2006]. De acordo com uma pesquisa realizada em [McDonald & Welland, 2001], a média de idade de um membro da equipe de desenvolvimento Web é aproximadamente 26 anos, e uma equipe de tamanho médio possui normalmente seis membros.

### 2.3. Processo de Desenvolvimento

Muitas características que são típicas para processos de desenvolvimento de aplicações Web podem também ser encontradas em processos de desenvolvimento de software tradicional. Os modelos de processo da Engenharia da Web adotam a filosofia do desenvolvimento ágil [Kappel et al., 2006], enfatizando, portanto, uma abordagem de desenvolvimento simples que incorpora ciclos rápidos de desenvolvimento. É importante observar que essa abordagem deve ser adaptável, isto é, deve ser capaz de acomodar possíveis mudanças nos requisitos, e deve permitir ainda que a aplicação seja desenvolvida paralelamente por vários sub-grupos da equipe de desenvolvimento.

Segundo Pressman (2006), imediatismo e evolução contínua são os principais atributos de uma aplicação Web, os quais podem afetar diretamente na escolha de um modelo de processo genérico. Assim qualquer modelo de processo da Engenharia da Web deve estar fixado em três pontos [Pressman, 2006] [Kappel et al., 2006]: entrega incremental, modificações contínuas, e cronogramas curtos. A Figura 3 ilustra um modelo de processo genérico, com base em [Pressman, 2006] e [Perrone et al., 2005b].

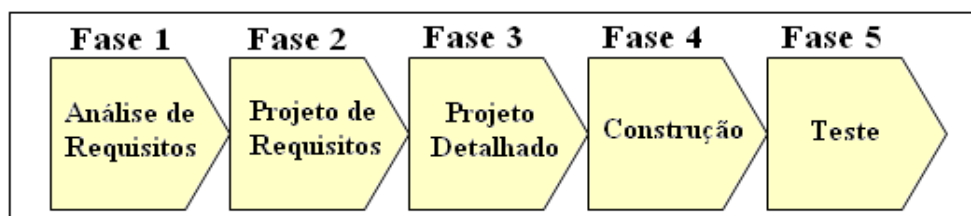


Figura 3: Modelo de Processo Genérico para aplicações Web. [Perrone et al., 2005b]

- **Fase 1:** essa fase compreende as atividades de concepção e levantamento de requisitos. O objetivo da atividade de concepção é estabelecer um

entendimento básico do problema, a natureza da solução desejada e a efetividade da comunicação e colaboração preliminares entre cliente e analistas [Pressman, 2006]. Durante o levantamento, todos os *stakeholders* devem ser identificados, requisitos do negócio são coletados e a integração entre a aplicação Web e outras aplicações de negócio, bancos de dados e funções são definidas.

- **Fase 2:** nesta fase, as informações obtidas de todos os *stakeholders* durante a concepção e o levantamento são expandidas e refinadas. Assim, o objetivo dessa fase é desenvolver um modelo técnico refinado das funções, características e restrições da aplicação Web, isto é, devem-se construir modelos de análise que representem que tipo de informação ou conteúdo deve ser apresentado, que funções devem ser realizadas pelo usuário final e a estrutura navegacional que será explorada à medida que o usuário interage com a aplicação.
- **Fase 3:** nessa fase, os modelos de análise construídos na fase anterior são refinados, e, além disso, são gerados modelos de projeto de interface e de componentes. O projeto de interface descreve a estrutura e organização da interface com o usuário, incluindo esquemas de cor, layout geométrico, tamanho, fonte e outras decisões estéticas relacionadas. O projeto de componentes desenvolve a lógica de processamento detalhada necessária para implementar os componentes funcionais.
- **Fase 4:** ferramentas e tecnologias da Engenharia da Web são aplicadas para construir a aplicação Web que foi modelada. Uma vasta gama de ferramentas e tecnologias tem evoluído desde a década passada à medida que as aplicações Web têm se tornado mais sofisticadas e difundidas.
- **Fase 5:** uma vez construída uma versão da aplicação Web, uma série de testes rápidos é conduzida para garantir que erros relacionados basicamente ao conteúdo, às funções e aos aspectos de navegação sejam descobertos.

Essas fases genéricas devem ocorrer repetidamente à medida que cada versão é submetida à engenharia e entregue. Modificações podem ocorrer como resultado da

avaliação de uma versão entregue ou como uma consequência de condições de negócio dinâmicas. Cada uma dessas fases deve ser conduzida em períodos curtos de tempo, diante das exigências imediatas do mercado.

## **2.4. Análise de Requisitos**

A ER, como todas as outras atividades de engenharia de software, precisa ser adaptada às necessidades do processo, do projeto, do produto e da equipe que está desempenhando o trabalho [Pressman, 2006]. No contexto de aplicações baseadas na Web, a ER se depara com novas necessidades que acomodem as características distintas desse tipo de aplicação.

Um dos desafios mais críticos enfrentados principalmente pela ER para a Web está relacionado à especificação inicial de requisitos insuficiente, desde que, na grande maioria das vezes, o desenvolvimento de uma aplicação Web é iniciado a partir de requisitos vagos ou mal-estruturados, fato que compromete todas as fases subsequentes do ciclo de desenvolvimento [Kappel et al., 2006]. De fato, a preocupação nessa fase deixa de estar relacionada ao “o que deve ser feito” para “quando deve ser feito” [Pressman, 2006].

Outro grande desafio é a necessidade de gerenciamento da complexidade de requisitos altamente voláteis, refletindo as exigências dinâmicas do mercado. Mudanças constantes nos requisitos e diversas restrições são características comuns de projetos Web. Assim, é fundamental que haja condições de gerenciar as interdependências entre requisitos, e entre requisitos e outros artefatos (rastreadabilidade) a fim de avaliar efetivamente o impacto de possíveis mudanças durante o desenvolvimento [Zhang & Chung, 2003].

O terceiro desafio diz respeito à necessidade de liberação de versões funcionais da aplicação Web logo no início do desenvolvimento. Dessa forma, os engenheiros Web precisam usar métodos de planejamento, análise, projeto, implementação e teste que tenham sido adaptados a prazos curtos, que podem variar de dias a algumas semanas. Outro problema está ligado ao gerenciamento de uma grande variedade de grupos de usuários cujos objetivos precisam ser satisfeitos pela aplicação a ser desenvolvida. Muitos stakeholders ainda são desconhecidos durante as atividades de ER e, além disso, é necessário, muitas vezes, levar em consideração requisitos de internautas e visitantes

ocasionais. Diante dessas circunstâncias, é fundamental encontrar representantes adequados que possam fornecer requisitos realísticos.

Outros fatores que devem ser tratados pela ER para a Web incluem aspectos de qualidade tais como segurança, alta disponibilidade, usabilidade, padrões de navegação consistentes, estética, previsibilidade de informação, escalabilidade, desempenho e qualidade do conteúdo, no que diz respeito ao seu nível de atualização, consistência e confiabilidade.

### 2.4.1. Métodos de Análise de Requisitos

O aumento da complexidade das aplicações Web e a necessidade cada vez maior de se obter produtos finais de alta qualidade são os principais fatores que justificam os esforços de muitos pesquisadores buscando estabelecer estratégias que dêem suporte efetivo à fase de ER para a Web.

Considerada como a primeira abordagem orientada a objetivos proposta na literatura, KAOS [Dardenne et al., 1993] é um método formal, adequado para a definição e análise de requisitos de sistemas de segurança crítica. O método é baseado na premissa de que para construir um software em um ambiente organizacional, é necessário analisar e modelar explicitamente esse ambiente, em termos de atores, seus objetivos e dependências. Uma avaliação sistemática da efetividade da metodologia Kaos assim como de uma ferramenta associada pode ser encontrada em [Al-Subaie & Maibaum, 2006]. Nesse contexto, o framework *i\** [Yu, 1993] introduz o conceito de agente organizacional na ER e propõe um conjunto de operadores para modelar dependências entre agentes com relação aos seus objetivos, tarefas e recursos a serem fornecidos.

Já a metodologia Tropos [Castro et al., 2002] compreende todas as atividades de análise e projeto durante o processo de desenvolvimento de software. O objetivo é construir um modelo tanto do sistema a ser desenvolvido quanto do seu ambiente, que deve ser refinado e estendido, de forma incremental, por meio de 4 fases: requisitos iniciais, requisitos bem elaborados, projeto arquitetural e projeto detalhado. A metodologia adota os mesmos conceitos oferecidos pelo *framework i\** como suporte às fases de requisitos.

GBRAM [Antón & Potts, 1998] é uma metodologia que combina a abordagem orientada a objetivos com técnicas baseadas em cenário. Além disso, o método estabelece um conjunto de heurísticas para identificar, analisar e refinar objetivos em requisitos. Assim como Tropos, essa metodologia não contempla os aspectos de navegação e de composição, inerentes aos artefatos de hipermídia como são as aplicações Web.

O método Vord foi estendido em [Al-Salem & Samaha, 2007] para atender às necessidades específicas de definição de requisitos para aplicações Web de *e-business*. Essa abordagem de ER tem como objetivo elicitar e formular requisitos relacionados a estratégias e processos de negócio em volta de diferentes pontos de vista, com ênfase nas operações e serviços que serão automatizados. Outros métodos de ER que incorporam estratégias e operações de negócios organizacionais para aplicações Web são encontrados em [Bleisten et al., 2004], [Gordijn & Akkermans, 2003] e [Meldrum & Rose, 2004].

MODFM [Zhang et al., 2003] [Zhang & Chung, 2003] [Zhang & Buy, 2003] é um método de prototipação rápida orientado a modelo, proposto para auxiliar nas atividades de elicitação e validação de requisitos para aplicações Web. O método aplica conceitos da ER tais como a técnica de casos de uso, análise estruturada, técnica de prototipação rápida e engenharia de requisitos baseada em arquitetura. Com base nesses conceitos e em uma proposta de arquitetura de software formalizada, o método estabelece um algoritmo que deve receber como entrada uma especificação funcional de alto-nível do sistema e gerar como saída um protótipo para a aprovação do cliente. A principal desvantagem do método MODFM é que sua efetividade depende fortemente da tecnologia utilizada no desenvolvimento. Além disso, a metodologia fornece suporte apenas para aplicações Web estritamente transacionais.

DSL [Redouane, 2002] é uma linguagem formal de especificação de requisitos para a Web, baseada no cálculo de predicado de primeira ordem. A especificação DSL é constituída de predicados e definições, divididos em 3 tipos principais para expressar a noção de tempo: estado, recurso e transição. O primeiro tipo – estado – é usado para representar eventos que podem ocorrer na aplicação (por exemplo, o clique de um botão). Já o segundo tipo – recurso – normalmente está relacionado a alguma entidade física na página (por exemplo, um menu), e o terceiro tipo – transição – refere-se a

tarefas desempenhadas pela própria aplicação (por exemplo, envio de alguma notificação ao usuário). Como em outros métodos formais, a grande limitação da DSL é a dificuldade de entendimento do formalismo matemático utilizado, conforme as especificações se tornam maiores e mais ricas. Um estudo de caso envolvendo esse método pode ser encontrado em [Redouane, 2004].

A seguir, será apresentado o modelo AWARE (*Analysis of Web Application Requirements*), adotado pelo método Web-SEMP para dar apoio às atividades de elicitação e especificação de requisitos para aplicações Web.

### 2.4.2. O Modelo AWARE

O modelo AWARE [Bolchini et al., 2003] [Bolchini & Paolini, 2004] adota uma abordagem semi-formal, orientada a objetivos para identificar e especificar requisitos de hipermídia, isto é, requisitos relacionados basicamente a aspectos de conteúdo, navegação, apresentação e interação de aplicações Web. Assim como em *i\** [Yu, 1993], o modelo AWARE permite elaborar e organizar objetivos estratégicos e de comunicação de diversos *stakeholders*, entretanto o modelo estende a análise de *i\** derivando requisitos de acordo com uma taxonomia proposta para ajudar a organizar a atividade de projeto.

A abordagem orientada a objetivos é especialmente adequada para o ambiente Web, uma vez que permite tratar objetivos vagos, imprecisos ou mal-definidos, refletindo necessidades de todos os *stakeholders* envolvidos no projeto [Bolchini & Mylopoulos, 2003]. Esses objetivos são tratados a partir da sua decomposição em objetivos menores – chamados de sub-objetivos – de forma que a compreensão das necessidades de um *stakeholder* se torne mais clara na medida que objetivos cada vez mais granulares vão sendo identificados. É importante observar que essa decomposição de objetivos fornece condições para definir tarefas e os primeiros requisitos de hipermídia.

Ao invés de prescrever um processo, o modelo AWARE fornece um conjunto de notações para modelar *stakeholders*, suas dependências com relação aos seus objetivos e tarefas, e requisitos específicos de hipermídia. As principais notações oferecidas pelo AWARE são:

- *Stakeholder*: Essa notação permite modelar cada perfil de usuário a ser considerado para a aplicação Web, além de todos os clientes relevantes e *stakeholders* envolvidos (representantes de companhia, gerentes de *marketing*, patrocinadores, etc.). Os *stakeholders* possuem seus próprios objetivos e podem depender de outros *stakeholders* para cumprir seus objetivos ou executar tarefas. Como um exemplo, o gerente de *marketing* de uma loja de compras on-line pode ter um objetivo relacionado ao ato de vender os produtos do patrocinador do site. No final do dia, o cumprimento desse objetivo dependerá, na verdade, das compras atuais realizadas pelos usuários do site.
- Prioridade de *Stakeholder* (*Stakeholder Priority*): Muitas vezes, é necessário decidir quais *stakeholders* são mais importantes do que outros, devido a razões organizacionais, ou devido a propósitos de comunicação ou de negócio. Um valor de prioridade pode então ser associado a cada *stakeholder* a fim de identificar e priorizar o público-alvo da aplicação, o que possibilita focar o esforço de comunicação e gastar os recursos de projeto com perfis de usuário específicos. A prioridade pode ser expressa por valores quantitativos (0,...,1) ou qualitativos (+,-,++,--,....).
- Objetivo (*Goal*): Um objetivo modela uma necessidade de alto nível de um ou mais *stakeholders*. Pode ser necessário que os objetivos sejam expressos informalmente ou semi-formalmente para tornar clara a comunicação entre analistas e *stakeholders*. Objetivos ainda podem ser decompostos em sub-objetivos, em uma maneira recursiva.
- Tarefa (*Task*): Refinando-se os objetivos ou sub-objetivos de usuários, são definidas as tarefas. Enquanto que um objetivo refere-se a um caso geral desejado pelo usuário, uma tarefa é uma atividade de alto nível realizada no site. No caso de aplicações Web voltadas ao paradigma transacional, as tarefas de usuário devem ser detalhadamente elaboradas para fornecer subsídios tanto à definição precisa de um conjunto de operações quanto à elaboração de uma estrutura de hipermídia adequada que possa dar suporte a essas tarefas.
- Prioridade de objetivo por *stakeholder* (*Stake Priority*): Um *stakeholder* pode considerar um objetivo mais importante do que outro, e, no caso de haver um

grande número de objetivos, seria recomendável “forçar” *stakeholders* a priorizar seus objetivos para fazer o projeto convergir a metas de aplicação cruciais. Os *stakeholders* e a equipe de projeto também podem concordar em considerar que alguns objetivos de usuários são mais improváveis de acontecer do que outros. Essa prioridade é então uma propriedade do relacionamento entre *stakeholders* e seus objetivos, sendo expressa por valores quantitativos (0,...,1) ou qualitativos (+,-,++,--,...).

- **Prioridade absoluta de objetivo (*Goal Priority*):** Como um objetivo pode ser possuído por mais de um *stakeholder*, a sua prioridade deve ser calculada adequadamente, combinando-se as prioridades de todos os *stakeholders* envolvidos e as prioridades que cada *stakeholder* atribuiu ao objetivo em questão. Por exemplo, os analistas podem multiplicar a prioridade de cada *stakeholder* envolvido no objetivo em questão pela sua prioridade atribuída a esse objetivo, somando, então, esses valores para obter a prioridade absoluta do objetivo.
- **Requisito (*Requirement*):** Os objetivos são refinados, transformados em sub-objetivos, tarefas e eventualmente em requisitos. Os requisitos de hipermídia são informalmente expressos em linguagem natural, e seu nível de detalhamento pode ser negociado entre analistas e a equipe de projeto. Os requisitos não têm como objetivo principal capturar todos os aspectos de hipermídia de uma aplicação Web, mas apenas características cruciais que sirvam de entrada para a atividade de projeto.

Para organizar o conjunto de requisitos de hipermídia, o modelo AWARE classifica os requisitos de acordo com o aspecto de projeto relacionado a eles. Essa taxonomia de requisitos compreende as seguintes dimensões:

- **Conteúdo (*Content*):** sendo o valor central de uma aplicação Web voltada ao paradigma de hipermídia, o conteúdo refere-se ao conjunto de idéias e mensagens que o site comunica aos seus usuários. No caso de um site de um museu, alguns requisitos de conteúdo poderiam ser: “apresentar detalhes de cada pintura”, “fornecer bibliografia para cada pintor”, “fornecer mensagens de boas-vindas ao visitante”, “informar hora de abertura e visita”, etc.



- Estrutura de conteúdo (*Structure of content*): os requisitos desse tipo têm como objetivo indicar como as partes do conteúdo identificado podem ser estruturadas. Especificar requisitos iniciais sobre a estrutura de conteúdo implica em expressar a necessidade de destacar partes de conteúdo ou mensagens dentro de um mesmo objeto de informação. No caso do exemplo do museu, tais requisitos poderiam ser: “na apresentação do museu, ressalte o valor histórico da construção”, ou, “apresentando a pintura, detalhe as técnicas usadas”, etc.
- Caminhos de acesso ao conteúdo (*Access paths to content*): essa dimensão de requisitos refere-se aos caminhos navegacionais disponíveis ao usuário para chegar ao conteúdo necessário. O usuário deve ter a permissão de acessar o conteúdo requerido ou explorar toda a informação oferecida pelo site seguindo os caminhos de acesso que melhor correspondem às suas expectativas ou objetivos. Portanto, essa dimensão tem por objetivo capturar as maneiras pelas quais o usuário poderá iniciar uma navegação, localizar e alcançar o conteúdo de interesse. Exemplos de tais requisitos para o site do museu poderiam ser: “permita acessar pinturas por autor”, “permita acessar autores por período e por nome”, “forneça acesso a trabalhos de arte recomendados”, “permita planejar uma visita por data e preferências”, etc.
- Navegação (*Navigation*): os requisitos dessa dimensão devem sugerir conexões entre diferentes partes de informação, permitindo que o usuário navegue de uma parte do conteúdo a outra semanticamente relacionada, completando, assim, sua tarefa cognitiva ou operacional. Essa dimensão de projeto captura os artefatos de hipermídia explorados pelo usuário para navegar a partir de um dado objeto de informação a um ou mais semanticamente relacionados. Exemplos de requisitos de navegação para o site do museu são: “relacione cada pintura ao seu autor”, “relacione informações de visitas a serviços de restaurantes e hotéis disponíveis”, etc.
- Apresentação (*Presentation*): os requisitos de apresentação estão relacionados às estratégias de comunicação visual para apresentar conteúdo, opções de navegação e operações ao usuário. Essa dimensão foca, na verdade, dois aspectos principais: elementos gráficos (botões, ícones, imagens, proporções de fontes, etc.) e leiaute de interface, que está relacionado ao posicionamento físico

desses objetos na página. Para o exemplo do museu, tais requisitos poderiam ser: “apresente um estilo jovem voltado a adolescentes na seção de crianças”, ou, “apresente um estilo profissional e artisticamente rico na seção de coleção de pinturas”, etc.

- Operação de usuário (*User operation*): são as operações visíveis aos usuários, os quais podem ativá-las para completar algumas tarefas interagindo com a aplicação. No exemplo do museu, alguns requisitos poderiam ser: “aderir a uma lista de *e-mail*”, “enviar comentários pessoais a uma pintura”, “criar coleção pessoal de pinturas”, etc.
- Operação de sistema (*System operation*): as operações de sistema não são diretamente visíveis aos usuários, mas são obrigatórias para construir as operações de usuário. Possíveis requisitos de operação de sistema incluem: “monitore a navegação do usuário”, “force a autenticação do usuário para criar sua coleção pessoal de pinturas”, etc.
- Interação (*Interaction*): os requisitos dessa dimensão descrevem os estilos de interação disponíveis ao usuário. É importante observar que esses requisitos estão intimamente relacionados aos aspectos de apresentação, entretanto, enquanto este último se preocupa com aspectos cognitivos e de estética das páginas Web, os requisitos de interação estão relacionados a recursos capazes de aumentar o interesse e envolvimento do usuário por algum assunto ou conteúdo oferecido pelo site. Um exemplo de requisito para o site do museu poderia ser: “forneça ao usuário um modelo 3D interativo de um trabalho especial de arte”.

Embora um requisito possa pertencer, inicialmente, a mais de uma dimensão ao mesmo tempo, o modelo AWARE sugere que esse requisito seja refinado até o ponto onde exatamente uma dimensão possa ser designada a ele. A Figura 4 ilustra o meta-modelo para a especificação de requisitos de uma aplicação Web.

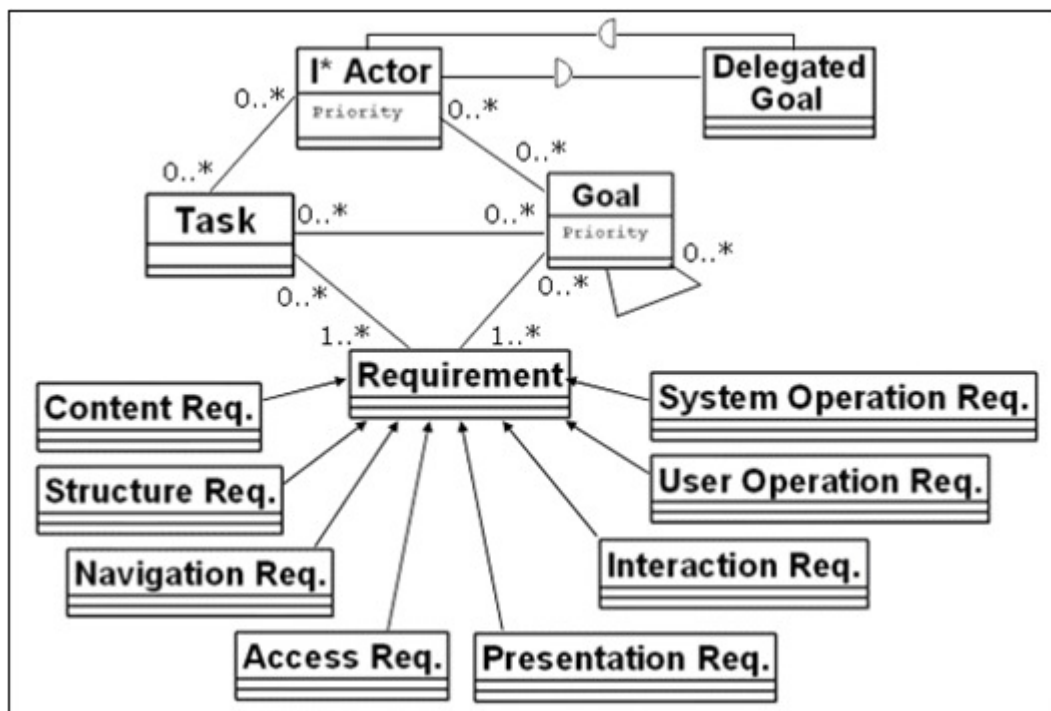


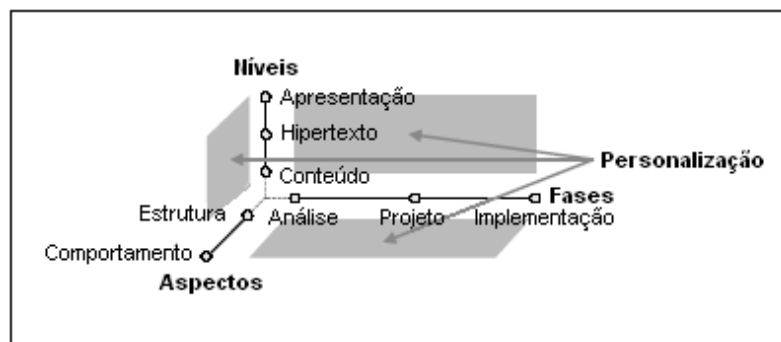
Figura 4: Meta-Modelo AWARE para aplicações Web. [Bolchini & Paolini, 2004]

Trechos de exemplos de aplicação do modelo AWARE na indústria podem ser encontrados em [Bolchini et al., 2003] e [Bolchini & Paolini, 2004], e experiências e lições aprendidas com a aplicação do modelo em outro estudo de caso real podem ser observadas em [Perrone & Bolchini, 2004] e [Perrone et al., 2005a].

## 2.5. Modelagem de Requisitos

Para dar continuidade ao processo de desenvolvimento de software, os requisitos levantados devem ser modelados por meio de uma técnica de modelagem, a qual pode ser formal, semi-formal ou informal. Para levar em consideração as características específicas de aplicações Web, a atividade de modelagem pode ser organizada de acordo com três dimensões [Kappel et al., 2006]: níveis, aspectos e fases, conforme mostra a Figura 5.

Há três níveis que devem ser distinguidos durante a atividade de modelagem: conteúdo, hipertexto e apresentação. Uma clara separação desses três níveis permite o reuso e contribui para reduzir a complexidade de desenvolvimento. Os aspectos estruturais e de comportamento devem ser modelados em cada um dos três níveis, conforme mostra a Figura 5. A relevância desses aspectos na modelagem depende do tipo de aplicação Web a ser desenvolvida – aplicação de hipermídia ou transacional.



**Figura 5: Domínios de Modelagem para Aplicações Web. [Adaptado de Kappel et al., 2006]**

As fases descritas na Figura 5 apenas mostram, de maneira simplificada, as etapas genéricas de processo discutidas anteriormente neste trabalho. A inclusão de informação de contexto – personalização – no desenvolvimento de aplicações Web desempenha um papel significativo no sentido de considerar preferências de usuários, características de dispositivos e restrições de largura de banda. Dessa forma, de acordo com a Figura 5, pode-se perceber que a personalização tem influência nos três níveis de modelagem (conteúdo, hipertexto e apresentação) e devem ser consideradas em todas as fases do processo de desenvolvimento.

A modelagem de conteúdo inclui a criação do modelo de domínio do problema, baseado em conceitos e métodos de modelagem de dados (ou modelagem orientada a objetos). Assim, esse modelo deve incluir aspectos estruturais do conteúdo e, dependendo do tipo de aplicação Web, aspectos comportamentais.

A modelagem de hipertexto, também conhecida como modelagem navegacional, permite especificar a navegabilidade através do conteúdo de uma aplicação Web, isto é, todos os caminhos navegacionais disponíveis aos usuários. A natureza não-linear de um hipertexto é uma das propriedades mais importantes que devem ser levadas em consideração durante a modelagem de aplicações Web, assim a estrutura de hipertexto deve ser projetada cuidadosamente.

O objetivo do modelo de hipertexto é, portanto, definir todas as categorias de informação que podem ser acessadas à medida que o usuário navega pela aplicação. Através desse modelo também é necessário especificar estruturas de acesso adequadas, tais como índices, menus, roteiros guiados ou consultas. Um índice é uma estrutura de

acesso que permite ao usuário selecionar um único objeto a partir de uma lista homogênea de objetos. Índices podem ser acrescentados automaticamente no modelo de hipertexto sempre que for necessário permitir o acesso a um conjunto (maior que 1) de objetos de um nodo.

Em contraste aos índices, menus são usados para que o usuário possa acessar nodos heterogêneos ou sub-menus. Roteiros guiados também são comuns em aplicações Web e permitem que o usuário navegue seqüencialmente por um certo número de nodos. Já uma consulta permite que o usuário encontre nodos que satisfaçam alguma condição específica, de acordo com os dados inseridos em um formulário. Estruturas de acesso recorrentes podem ser descritas em termos de padrões de projeto ou padrões de navegação.

A maioria dos métodos de modelagem para a Web oferece elementos voltados para os padrões de navegação mais freqüentemente usados. Alguns desses padrões especiais incluem a notação “*Home*”, que aponta para a página *home* de uma aplicação Web, e a notação “*Landmark*”, que se refere a um nodo que pode ser alcançado por meio de qualquer outro nodo da aplicação.

A modelagem de apresentação está relacionada às estratégias de comunicação visual utilizadas e formas de apresentação de conteúdo e funcionalidades aos usuários. Assim, o elemento-chave desse nível é a interface de usuário, a qual deve ser simples e fácil de ser entendida de modo a evitar a desorientação do usuário.

### 2.5.1. Métodos de Modelagem

Para permitir a modelagem nos três níveis explicados anteriormente – conteúdo, hipertexto e apresentação –, uma grande variedade de métodos de projeto para a Web tem sido proposta na literatura, nas últimas décadas. HDM [Garzoto & Paolini, 1993] foi o primeiro método de projeto orientado a modelo e influenciou algumas propostas subsequentes tais como W2000 [Baresi et al., 2001] [Baresi et al., 2006], OOHDM [Rossi et al., 2001] e WebML [Ceri et al., 2000] [Ceri et al., 2002].

W2000 é uma abordagem de projeto conceitual orientada a modelo que adota a separação de interesses por meio de quatro etapas: projeto de informação, projeto de navegação, projeto de apresentação e projeto de operação. O projeto de informação

define os dados usados pela aplicação e observados pelo usuário; o projeto de navegação e operação define como o usuário pode navegar pelo espaço de informação e modificá-lo através de processos de negócio, e o projeto de apresentação define como dados e serviços são apresentados ao usuário.

Outro método influenciado pelo HDM, o OOHDH é um modelo de projeto orientado a objetos que permite a especificação de aplicações Web de hipermídia, por meio de quatro atividades: modelagem conceitual, projeto da navegação, projeto da interface abstrata e implementação. Durante a modelagem conceitual, um modelo do domínio da aplicação é elaborado utilizando-se princípios bem conhecidos de modelagem orientada a objetos, acrescentados de algumas primitivas tais como perspectivas de atributos e sub-sistemas.

Durante o projeto navegacional, diferentes visões navegacionais são construídas a partir de um único modelo conceitual, levando-se em consideração os tipos de usuários aos quais a aplicação se destina e o conjunto de tarefas que deverão desempenhar. A estrutura navegacional é descrita em termos de nodos – mapeados a partir de classes conceituais –, e elos – mapeados a partir de relacionamentos do modelo conceitual. No projeto navegacional, é definida também a maneira como a navegação irá proceder, especificando transformações no espaço navegacional, isto é, o conjunto de objetos navegacionais acessíveis em cada momento. O projeto da interface abstrata define um modelo especificando quais objetos de interface serão vistos pelo usuário, a forma que tomarão diferentes objetos navegacionais e quais transformações ocorrerão na interface. Tanto o projeto quanto a geração de código podem ser auxiliados por uma ferramenta CASE chamada OOHDH-Web.

Conallen (2003) propôs uma extensão da UML utilizando um modelo orientado a objetos em conjunto com a abordagem de casos de uso como base para a modelagem de aplicações Web. De acordo com essa proposta, a UML é estendida através da inclusão de novos estereótipos na linguagem possibilitando tratar basicamente páginas, *links* e formulários, assim como as associações entre esses elementos e as classes do modelo de domínio do problema.

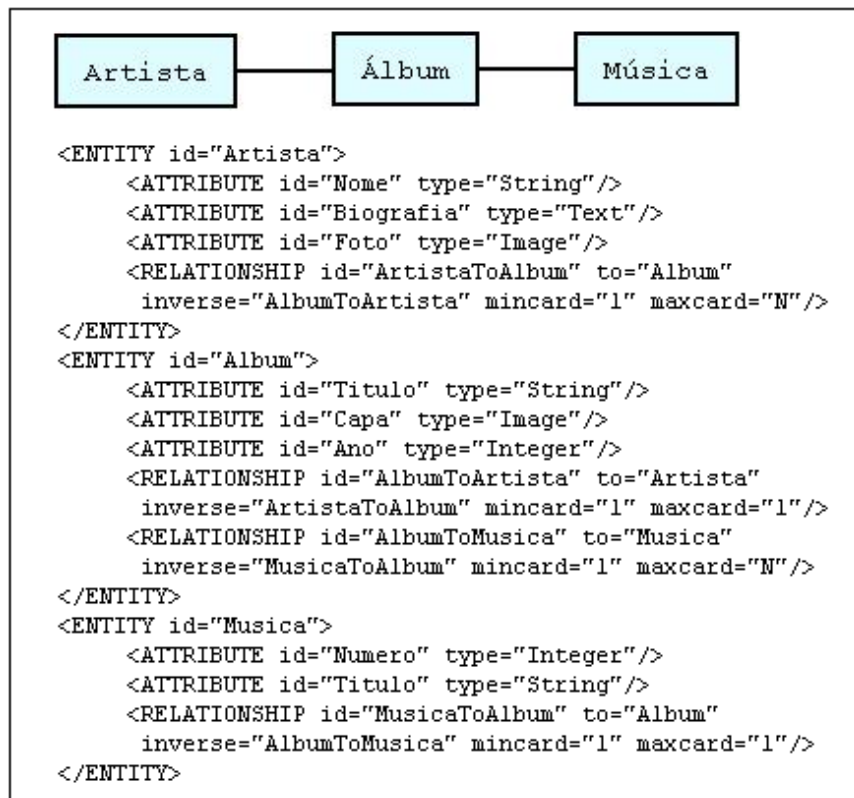
### 2.5.2. WebML

WebML (Web Modeling Language) [Ceri et al., 2000] [Ceri et al., 2002] é uma linguagem visual para especificar aplicações Web complexas em um nível conceitual adequado, isto é, sem que haja comprometimento com detalhes arquiteturais e de implementação. Todos os conceitos da WebML são associados a uma notação gráfica e uma sintaxe em XML que, por sua vez, pode servir de entrada para a geração de código automática.

A especificação de uma aplicação Web em WebML consiste de quatro dimensões ortogonais: modelo estrutural (modelo de dados), modelo de hipertexto, constituído pelos sub-modelos de navegação e de composição, modelo de apresentação e modelo de personalização. Entretanto, o projeto de uma aplicação Web é baseado essencialmente em duas perspectivas ortogonais: especificação de dados e de navegação, contemplados, respectivamente, pelo modelo estrutural e pelo modelo de hipertexto, conforme é descrito a seguir.

#### Modelo Estrutural

O modelo estrutural expressa o conteúdo exibido e mantido pela aplicação Web, em termos de entidades e relacionamentos relevantes. Entidades são definidas como objetos de dados, e relacionamentos indicam conexões semânticas entre entidades. Entidades possuem atributos com um tipo associado e podem ser organizadas em hierarquias de generalização, e os relacionamentos possuem papéis e são caracterizados especificando-se as restrições de cardinalidade. A modelagem de dados em WebML é baseada no Modelo Entidade-Relacionamento (ME-R) e é compatível com o diagrama de classes da UML, no caso de uma abordagem orientada a objetos (OO). A Figura 6 mostra um exemplo do modelo estrutural e o código XML associado a esse modelo.



**Figura 6: Exemplo de Modelo Estrutural e Código XML.**

### Modelo de Hipertexto

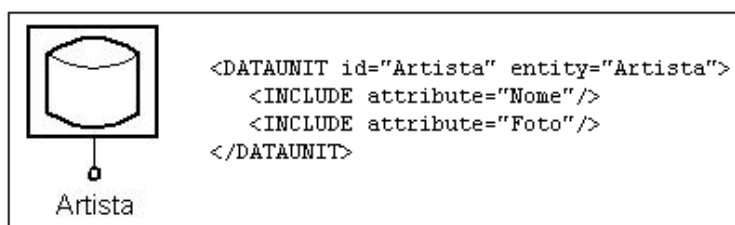
O modelo de hipertexto descreve um ou mais hipertextos que podem ser publicados na aplicação Web, e é composto pelos sub-modelos de composição e navegação, conforme já mencionado. A estrutura geral do hipertexto é definida em termos de *site views* (visões), áreas, páginas e unidades de conteúdo. Uma *site view* é um hipertexto, definida para satisfazer um conjunto específico de requisitos, assim, várias *site views* podem ser identificadas a partir do mesmo modelo estrutural. A descrição de uma *site view* consiste dos sub-modelos de composição e navegação. O primeiro sub-modelo especifica quais páginas compõem o hipertexto e quais unidades de conteúdo constituem uma página. Já o segundo sub-modelo expressa como páginas e unidades de conteúdo são ligadas para formar o hipertexto.

Uma unidade de conteúdo é um elemento atômico de publicação de informação na aplicação Web. A WebML provê cinco tipos de unidades de conteúdo para compor um hipertexto. São elas: *Data Units*, *Multidata Units*, *Index Units*, *Scroller Units* e *Entry Units*. A especificação de uma unidade de conteúdo (com exceção da *Entry Unit*) requer a definição de uma “fonte” e um “selecionador”. A fonte é o nome da entidade



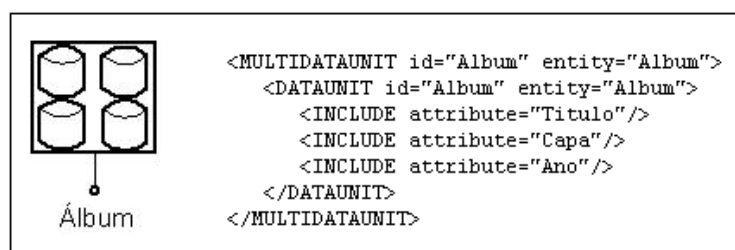
(ou classe em OO) a partir da qual o conteúdo publicado pela unidade é extraído; o selecionador é uma condição, usado para recuperar instâncias (ou objetos) da entidade fonte que satisfaçam a condição.

Uma *data unit* (unidade de dados) é definida para publicar informações acerca de uma única instância de uma dada entidade no modelo estrutural. Mais de uma *data unit* pode ser definida para a mesma entidade a fim de oferecer pontos de vista alternativos (diferentes perspectivas de atributos). A Figura 7 ilustra a notação gráfica para uma *data unit* e o código XML correspondente. A especificação de uma *data unit*, em XML, requer a indicação da entidade correspondente e a seleção dos atributos a incluir. O elemento DATAUNIT é usado para definir cada *data unit* e o elemento INCLUDE especifica quais atributos devem ser exibidos.



**Figura 7: Notação Gráfica e Código XML para uma Data Unit.**

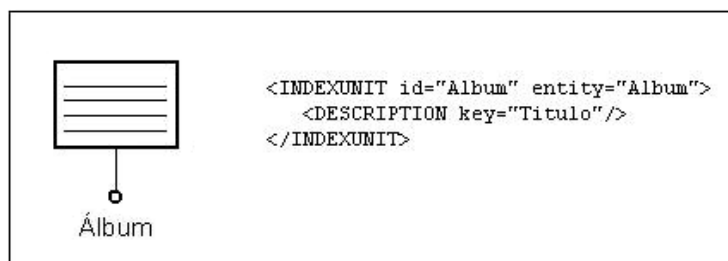
Uma *multidata unit* exibe múltiplas instâncias de uma entidade ao mesmo tempo, repetindo a apresentação de várias *data units* idênticas. Portanto, a especificação de uma *multidata unit* requer a identificação da entidade cujas instâncias serão apresentadas e a identificação da *data unit* usada para a apresentação de cada instância da entidade, conforme ilustra a Figura 8. Uma *multidata unit* é representada pelo elemento MULTIDATAUNIT, que inclui o elemento DATAUNIT aninhado.



**Figura 8: Notação Gráfica e Código XML para uma MultiData Unit.**

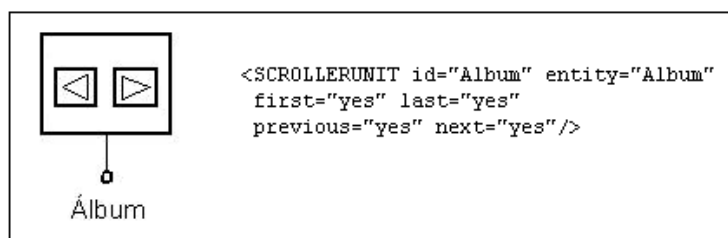
*Index units* são usadas para exibir múltiplas instâncias de uma entidade por meio de uma lista, denotando cada instância como uma entrada da lista (índice). De acordo

com a Figura 9, a especificação de uma *index unit* requer a identificação da entidade cujas instâncias serão exibidas e a identificação dos atributos que serão usados como a chave do índice. O elemento INDEXUNIT é usado e contém um elemento DESCRIPTION aninhado, conforme ilustra a Figura 9. *Index Units* são consideradas como estruturas de acesso, permitindo a navegação entre partes de conteúdo em um hipertexto.



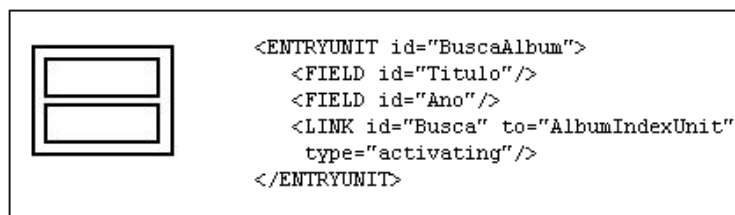
**Figura 9: Notação Gráfica e Código XML para uma *Index Unit*.**

*Scroller units* fornecem comandos que permitem a navegação entre instâncias de uma entidade. Uma *scroller unit* é usada em conjunção com *data units*, *multidata units* e *index units*. Sintaticamente, uma *scroller unit* é especificada usando-se o elemento SCROLLERUNIT, conforme é mostrado pela Figura 10.



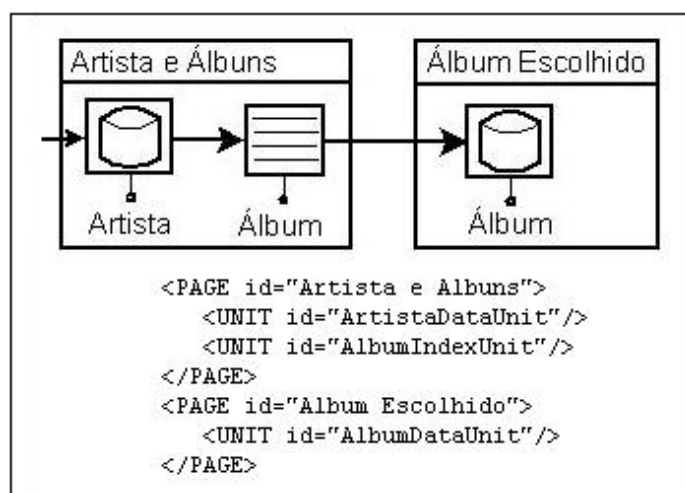
**Figura 10: Notação Gráfica e Código XML para uma *Scroller Unit*.**

*Entry units* são usadas para coletar valores de entrada que, uma vez submetidos, podem então ser enviados como parâmetros para outras unidades de conteúdo ou operações (por exemplo, operações de atualização de conteúdo, buscas, etc.). A especificação de uma *entry unit* requer a identificação de cada campo cujo valor será inserido pelo usuário e a identificação da unidade de destino (unidade de conteúdo ou unidade de operação). O elemento ENTRYUNIT é usado para definir cada *entry unit* e o elemento FIELD especifica quais campos devem ser exibidos, conforma mostra a Figura 11.



**Figura 11: Notação Gráfica e Código XML para uma *Entry Unit*.**

A fim de possibilitar que informações contidas em várias unidades de conteúdo sejam apresentadas juntas (por exemplo, dados de um artista e o índice de álbuns publicados por ele), a WebML fornece a noção de página (*page*). Uma página é a abstração de uma região da tela, constituída de unidades de conteúdo ou de outras páginas, em uma maneira recursiva. Sub-páginas aninhadas podem estar na forma conjuntiva (sub-páginas exibidas juntas) ou na forma disjuntiva (apenas uma sub-página é exibida ao mesmo tempo). Sintaticamente o elemento PAGE deve ser usado para definir cada página da aplicação Web, conforme ilustra a Figura 12.

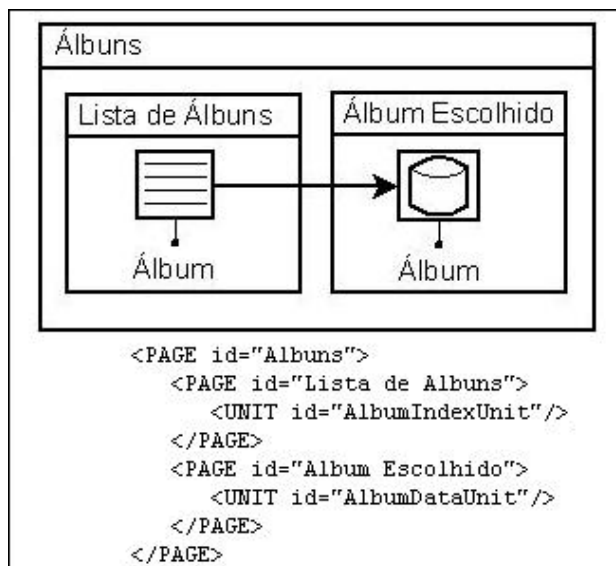


**Figura 12: Notação Gráfica e Código XML para uma Página.**

De acordo com a figura acima, a página “Artista e Álbuns” apresenta informações sobre um artista específico, publicadas por uma *data unit*, e a lista correspondente de álbuns, publicada por uma *index unit*. A página “Álbum Escolhido” apresenta informações específicas sobre o álbum selecionado pelo usuário na página anterior.

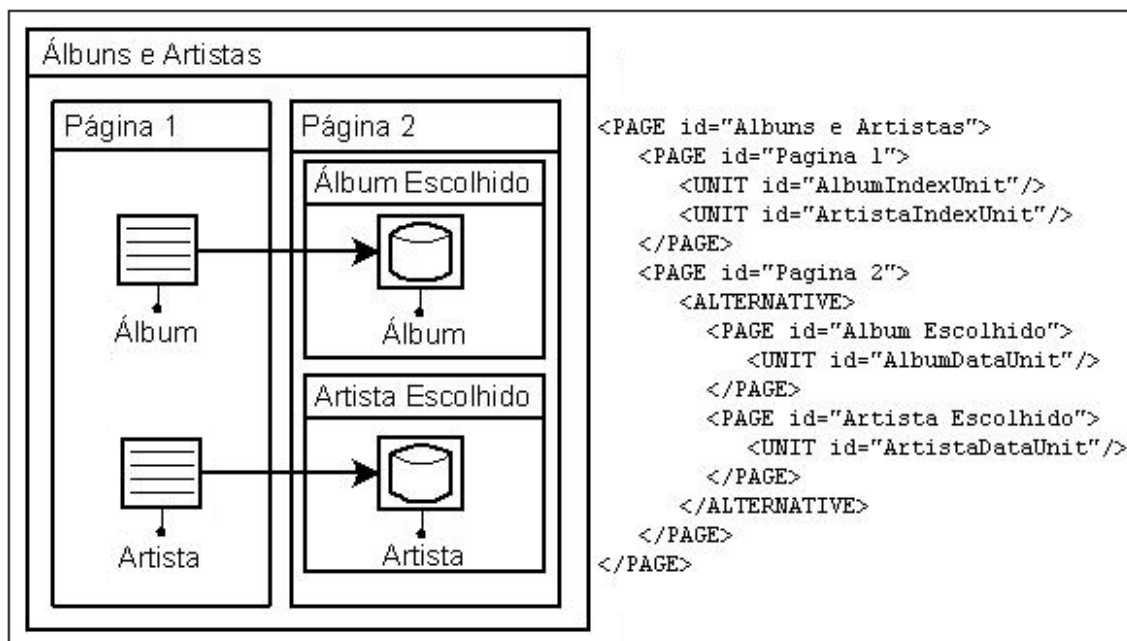
A Figura 13 representa duas sub-páginas aninhadas na forma AND (forma conjuntiva). De acordo com a figura, pode-se observar que o índice de álbuns continua

sendo exibido após o usuário ter escolhido uma opção da lista, isto é, uma porção da página se mantém fixa enquanto que a outra porção contém informações que podem variar de acordo com as ações do usuário.



**Figura 13: Notação Gráfica e Código XML para Sub-Páginas AND.**

Na Figura 14, pode-se observar o aninhamento de duas sub-páginas na forma OR (forma disjuntiva). Sub-páginas na forma OR são especificadas sintaticamente usando-se o elemento ALTERNATIVE, conforme ilustra a Figura 14. No exemplo dado, existem dois índices distintos em uma página (índice de artistas e álbuns) e cada um está relacionado a uma unidade de conteúdo diferente, cada qual contida em uma página. Assim, essas duas páginas devem ser estruturadas na forma OR, expressando a idéia de que apenas uma delas pode ser exibida ao mesmo tempo. É importante ressaltar que a especificação de páginas na forma AND/OR permite representar muitas estruturas de página complexas comumente identificadas em aplicações Web.



**Figura 14: Notação Gráfica e Código XML para Sub-Páginas OR.**

Conforme já mencionado, unidades de conteúdo e páginas devem estar interconectadas por meio de *links* para caracterizar um *site view*. Semanticamente, existem duas categorias de *links*: *links* contextuais e não-contextuais. *Links* contextuais conectam duas unidades de conteúdo, carregando alguma informação (contexto) que a unidade de destino deverá usar a fim de selecionar uma instância ou um conjunto de instâncias dos dados a serem exibidos. *Links* contextuais ainda podem ser divididos em *links* normais, quando há intervenção do usuário no que diz respeito à informação carregada pelo *link*, e *links* de transporte, quando não há intervenção do usuário. *Links* contextuais são denotados, em XML, pelo elemento INFOLINK, aninhado dentro de unidades e páginas. A Figura 15 exemplifica o uso de *links* contextuais.

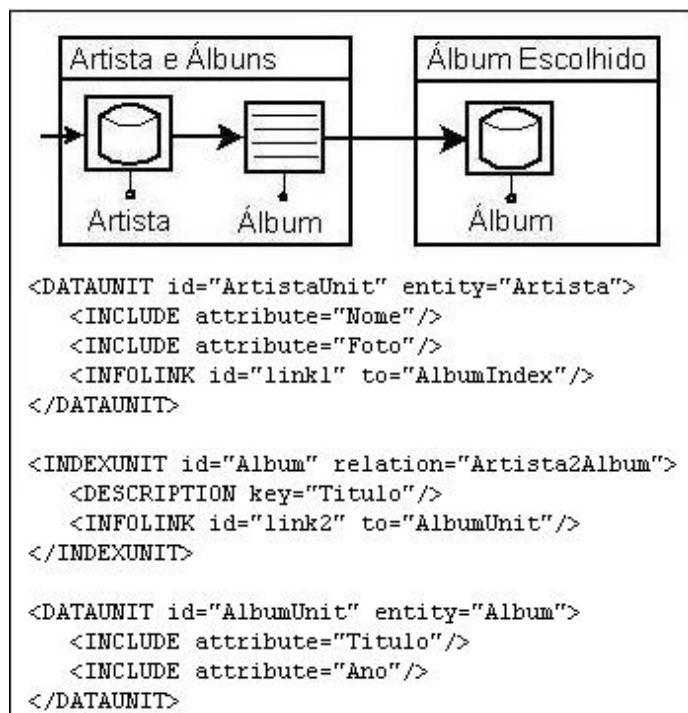


Figura 15: Notação Gráfica e Código XML para *Links* Contextuais.

*Links* não-contextuais conectam páginas, independentemente das unidades que as constituem e das relações semânticas entre conceitos estruturais incluídos nessas unidades. Sintaticamente, *links* não-contextuais são denotados pelo elemento HYPERLINK, aninhado dentro de páginas, conforme ilustra a Figura 16.

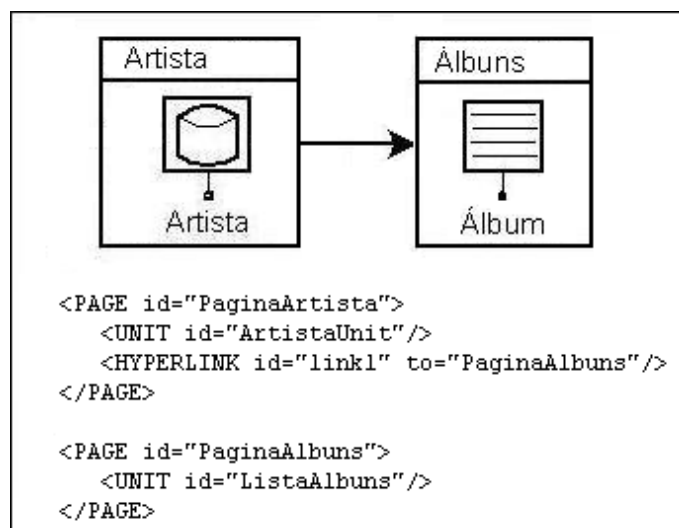


Figura 16: Notação Gráfica e Código XML para *Links* Não-Contextuais.

A WebML ainda dispõe de unidades de operações pré-definidas de gerenciamento de conteúdo que permitem criar, remover e alterar instâncias de uma

entidade, além de adicionar ou destruir um relacionamento entre duas instâncias. Unidades de operação possuem um número fixo de parâmetros de entrada carregados por meio de um ou dois *links* contextuais, mas um número arbitrário de parâmetros de saída. Dois de todos os links de saída de uma unidade de operação devem ser obrigatoriamente do tipo OK e KO. O primeiro tipo – tipo OK – é ativado caso a operação referenciada pela unidade tenha sido efetuada com sucesso; já o segundo tipo – tipo KO – é apenas ativado caso haja alguma falha durante a operação.

O modelo de apresentação da WebML expressa o leiaute e a aparência gráfica das páginas (fontes, cores, margens, etc.) por meio de uma sintaxe XML abstrata. A WebML também permite a personalização em termos de conteúdo entregue aos usuários, modelando explicitamente no esquema estrutural usuários e grupos na forma de entidades pré-definidas.

Em [Ceri et al., 2003], a WebML foi estendida para permitir representar mais amplamente em nível de dados noções de contexto, em termos de propriedades relacionadas ao usuário corrente, suas atividades, sua localização e dispositivos usados. O objetivo é fornecer conteúdo e serviços personalizados introduzindo um sub-esquema de dados para representar o modelo de contexto e especificando operações em nível de hipertexto. Em [Martino et al., 2007], outra proposta estendeu a WebML para modelar sistemas de informações geográficas baseados na Web. A extensão é baseada na necessidade de modelar tarefas de usuários diretamente sobre um mapa; tais tarefas geralmente produzem dados que não são apresentados em uma nova página, mas são renderizados no mesmo mapa, exigindo, portanto, novas unidades de conteúdo WebML.

### **2.5.3. Modelo de Casos de Uso Funcionais e Navegacionais**

O Modelo de Casos de Uso é outra técnica de modelagem de requisitos amplamente utilizada na indústria. O conceito de caso de uso foi proposto por Jacobson et al. [1992] em um modelo de processo denominado *Objectory*. Em essência, um caso de uso descreve seqüências de ações que um sistema executa produzindo um resultado de valor observável para um ator [Leffingwell & Widrig, 2003]. De acordo com [Pressman, 2006], atores são os diferentes indivíduos (ou dispositivos) que usam o sistema dentro do contexto da função e do comportamento que devem ser descritos.

Seqüências de ações podem descrever um conjunto de funções executadas, um procedimento algorítmico ou qualquer outro processo interno que produz algum resultado [Leffingwell & Widrig, 2003]. Uma seqüência ou cenário é invocado quando o ator inicia o caso de uso fornecendo alguma entrada ao sistema. Dessa forma, há um cenário básico e vários cenários alternativos, os quais são executados apenas sob certas circunstâncias.

Casos de uso podem possuir dois tipos de associações: extensão (representada pelo estereótipo <<extend>>) e inclusão (representada pelo estereótipo <<include>>). O primeiro tipo de associação é usado para estender o comportamento de um caso de uso existente; já o segundo tipo de associação é usado para evitar a duplicação de passos através de múltiplos casos de uso.

Para dar suporte às características específicas de aplicações Web, foi proposta a divisão do modelo de casos de uso em duas categorias [Baresi et al., 2001] [Koch & Kraus, 2002]: casos de uso funcionais e casos de uso navegacionais. Nessa abordagem, os casos de uso funcionais representam funcionalidades e serviços providos pela aplicação, da mesma forma que em sistemas de informação convencionais. Já os casos de uso navegacionais descrevem todos os caminhos que podem ser percorridos pelo usuário para alcançar conteúdos específicos na aplicação Web, e, portanto, são apoiados pelo modelo de hipertexto da aplicação.

Em [Baresi et al., 2001], sugerem-se criar dois modelos de casos de uso distintos, e em [Koch & Kraus, 2002] é proposta a criação de um único modelo de casos de uso, que usa o estereótipo <<navigation>> para denotar a diferença entre os casos de uso funcionais e aqueles que são específicos de hipertexto. Esta última alternativa é a adotada neste trabalho.

De acordo com [Escalona & Koch, 2006], um caso de uso navegacional compreende um conjunto de atividades de navegação que o usuário final poderá realizar para alcançar um nodo-alvo. Ainda de acordo com esses autores, uma atividade de navegação pode ser definida como a ação de seguir um *link* na aplicação Web, assim, de acordo com essa definição, cada caso de uso navegacional pode ser descrito como um conjunto de caminhos navegacionais diferentes disponíveis aos usuários.



## 2.6. Planejamento

Um dos aspectos cruciais do planejamento e gerenciamento de projetos é a compreensão de quanto o software provavelmente custará. Estimativas de custo elevadas podem fazer com que os clientes cancelem projetos, ao passo que uma estimativa de custo abaixo do real pode forçar a equipe do projeto a investir muito do seu tempo, sem que haja compensação financeira [Pfleeger, 2004].

No contexto da Engenharia da Web, assim como na Engenharia de Software tradicional, as métricas têm três objetivos principais [Pressman, 2006]: (1) fornecer uma indicação da qualidade da aplicação de um ponto de vista técnico; (2) fornecer uma base para estimativas de esforço e custo mais próximas da realidade; e (3) fornecer uma indicação da aplicação a ser desenvolvida do ponto de vista do negócio.

Focando no segundo objetivo, Mendes et al. (2001) definiram alguns fatores que dificultam o processo de estimativa para o desenvolvimento Web. Um desses fatores diz respeito à ausência de abordagens de modelagem de análise e projeto que possibilitem a aplicação de métricas que dêem suporte efetivo às atividades de planejamento e controle do projeto. Mendes et al. (2001) também ressaltam que os membros da equipe de Engenharia da Web são representados, na maioria das vezes, por projetistas e desenvolvedores menos experientes, conforme comentado neste capítulo. Os membros de uma equipe de desenvolvimento Web normalmente estão envolvidos em projetos cujo objetivo principal é colocar aplicações de alta qualidade no mercado tão rapidamente quanto possível [Reifer, 2000].

Nas próximas sub-seções são apresentados alguns dos principais métodos para planejamento de aplicações Web propostos na literatura. A primeira sub-seção discute alguns métodos propostos para avaliar a complexidade de aspectos de hipermídia de aplicações Web, e a segunda sub-seção discute métodos e trabalhos relacionados à avaliação de complexidade de aspectos funcionais de uma aplicação Web.

### 2.6.1. Métodos para Planejamento de Hipermídia

Proposto por Cleary (2000), o método *Web Points* é um método para avaliar o tamanho de sites Web estáticos, baseando-se no número e na complexidade de páginas HTML. Neste método, a complexidade de cada página é classificada em três níveis (baixa, média e alta) e um determinado valor de pontos Web é atribuído a cada página,

de acordo com dois fatores: o tamanho da página em palavras, e o número combinado de *links* internos e externos somado ao número de elementos não-textuais contidos na página. A Tabela 1 mostra os níveis de complexidade associados a uma página, de acordo com esses fatores, e a Tabela 2 mostra o número de pontos Web designado a cada nível de complexidade.

**Tabela 1: Níveis de Complexidade para uma Página em Web Points. [Cleary, 2000]**

Complexidade de Página HTML				
Palavras	Links Internos e Externos e Elementos Não-Textuais			
		0-5	6-15	>15
	0-300	Baixa	Baixa	Média
	301-500	Baixa	Média	Alta
	>500	Média	Alta	Alta

**Tabela 2: Pontos Web para cada Nível de Complexidade. [Cleary, 2000]**

Tamanho da Página HTML	
Complexidade	Pontos Web
Baixa	4
Média	6
Alta	7

O cálculo do valor total de pontos Web para a aplicação é dado pelo somatório de pontos atribuídos a cada página. É importante ressaltar que a estimativa de esforço estabelecida por esse método inclui as atividades de análise de requisitos, projeto, construção, teste e implementação de páginas HTML, portanto as atividades de criação e obtenção do conteúdo da página não são consideradas pelo método.

Buscando avaliar o tamanho de sites Web estáticos, Mendes et al. (2001) realizaram um estudo de caso que envolveu 43 estudantes de um curso de graduação a fim de coletar, especificamente, diversas métricas de tamanho e fatores de custo a serem usados como parâmetros na estimativa de esforço envolvido no projeto, criação e publicação de aplicações Web. Os dados coletados foram usados para gerar modelos de estimativas de esforço, baseados em duas técnicas estatísticas – regressão linear e regressão múltipla. Além disso, as métricas foram classificadas em três categorias: métricas de extensão, complexidade estrutural e funcionalidade [Mendes et al., 2001] [Mendes et al., 2002].

Em [Mendes et al., 2003] e [Mendes et al., 2005a] foram conduzidas duas pesquisas e um estudo de caso com o intuito de identificar novas métricas de tamanho e fatores de custo para aplicações Web, com base em práticas atuais de várias companhias de desenvolvimento Web pelo mundo inteiro. A primeira pesquisa usou uma máquina de busca para obter formulários de orçamento on-line de projetos fornecidos pelas companhias. Todas as métricas foram organizadas em categorias e classificadas, e os resultados obtidos indicaram que, dentre as métricas relacionadas aos aspectos de hipermídia, as quatro métricas de tamanho mais comumente usadas para estimativas foram: 1- número total de páginas Web (70%), 2- número total de imagens fornecidas pelo cliente” (31%), 3- número total de textos fornecidos pelo cliente” (30%) e 4- número total de fotos fornecidas pelo cliente.

Na segunda pesquisa, os resultados obtidos foram validados por meio de uma entrevista conduzida em uma companhia de desenvolvimento Web no Brasil. Com base no estudo de caso realizado, os resultados foram validados pela segunda vez, de acordo com uma pesquisa realizada em 32 companhias Web da Nova Zelândia.

Em [Mendes et al., 2005b] revisaram-se algumas das principais métricas para avaliar o tamanho de aplicações Web de hipermídia propostas na literatura nos últimos 12 anos. Com base nesse estudo, Mendes et al. (2005b) propuseram uma taxonomia para classificar as métricas pesquisadas e organizar conceitos de medida de software usados por cada métrica analisada.

### **2.6.2. Métodos para Planejamento Funcional**

Em 1998, a organização IFPUG (1998) (*International Function Point Users Group*) publicou diretrizes para medir aplicações Web usando as mesmas regras de contagem definidas no método Pontos por Função (PF) [Albrecht 1979]. Rollo (2000), no entanto, identificou dificuldades e contradições entre tais diretrizes no contexto de aplicações Web, e verificou que alguns aspectos importantes de contagem não eram considerados.

De fato, métodos propostos para medir o tamanho funcional tais como PF [Albrecht, 1979] [Albrecht & Gaffney, 1983] e Pontos por Casos de Uso (PCU) [Karner, 1993] foram elaborados em uma era “pré-Web” e, portanto, não foram projetados para levar em consideração os aspectos de hipermídia de tais aplicações.

Nesse contexto, é importante observar que métodos que avaliem o tamanho funcional devem ser utilizados em conjunto com algum método específico para avaliar aspectos inerentes às aplicações Web.

Reifer (2000) propôs o método Web Objects, definindo quatro componentes relacionados a aplicações Web (arquivos de multimídia, blocos de construção tais como ActiveX, DCOM, *applets*, etc., *scripts* e *links*) para que sejam usados juntamente com os componentes lógicos tradicionais do método PF (entradas externas, saídas externas, consultas externas, arquivos lógicos internos e arquivos de interface externa). O grande benefício desse método é o uso de uma base matemática sólida para medir uma aplicação Web, entretanto a sua maior limitação é a dependência de tecnologia de implementação, o que torna difícil sua aplicação durante as fases iniciais de análise de requisitos.

Em [Ruhe et al., 2003], Ruhe et al. compararam, através de regressão multivariável, o método Web Objects com o método PF, para estimar o esforço de desenvolvimento de aplicações Web. Os resultados da análise empírica revelaram que o método Web Objects produziu estimativas de esforço mais adequadas em relação ao método PF.

Em [Carvalho et al., 2006], o método PF foi aplicada no desenvolvimento de uma aplicação Web real com o intuito de avaliar seu grau de adequação a este tipo de ambiente. Outros dois métodos – Web Points e PCU – foram aplicados ao mesmo problema, e os resultados mostraram que o método PF, dentre os três analisados, foi o que permitiu alcançar um valor de esforço mais próximo do valor real consumido. Entretanto, é importante considerar que a aplicação Web utilizada no estudo de caso envolvia poucos componentes de multimídia, portanto aspectos inerentes ao ambiente Web não foram contabilizados pelas métricas analisadas.

O método PCU foi adotado neste trabalho para avaliar a complexidade funcional de aplicações Web por dois motivos: é um método simples, se comparado a outros métodos de planejamento, e é apoiado pelo Modelo de Casos de Uso, o qual é amplamente utilizado na indústria, conforme já comentado. De acordo com o método PCU, o cálculo de esforço é realizado com base na complexidade dos casos de uso que compõem o sistema bem como na complexidade dos atores que irão utilizá-lo. Esse

cálculo ainda leva em consideração os fatores de complexidade técnica cobrindo uma série de requisitos não-funcionais do sistema, e fatores ambientais que avaliam o nível de competência da equipe de desenvolvimento. Para fazer o mapeamento entre pontos e horas, Karner (1993) propôs um fator de 20 homens-hora por cada ponto de caso de uso, entretanto vários pesquisadores têm mostrado que essa conversão deve ser baseada no nível de experiência da equipe de desenvolvimento.

Em [Sanches et al., 2007], o método PCU foi utilizado em conjunto com o método PSP para ajustar o valor de pontos por casos de uso de acordo com dados provenientes da aplicação do PSP, isto é, o valor de pontos obtido por meio do método PCU é convertido em horas de acordo com características de cada membro da equipe de desenvolvimento. De acordo com esse trabalho, o valor do nível de esforço de cada membro da equipe é referenciado como NDE (Nível De Esforço).

## **2.7. Considerações Finais**

Neste capítulo foram apresentados os principais conceitos e trabalhos relacionados ao tema da dissertação. Assim, comentaram-se as principais características de aplicações Web que permitem distingui-las de aplicações convencionais, exigindo que conceitos e princípios da engenharia de software tradicional sejam adaptados de alguma forma para tratar tais características. Também foram discutidas as atividades de análise e modelagem de requisitos, e planejamento para aplicações Web, uma vez que estas são as atividades compreendidas pelo método apresentado neste trabalho. Para cada atividade, comentaram-se conceitos importantes e trabalhos relacionados, os quais deram subsídios à elaboração tanto da métrica WHP quanto do método Web-SEMP, apresentados, respectivamente, nos Capítulos 4 e 5.

# CAPÍTULO 3

## DEFINIÇÃO DA MÉTRICA WHP

---

### **3.1. Considerações Iniciais**

Embora existam algumas métricas para avaliar o tamanho de aplicações Web propostas na literatura, conforme citado no Capítulo 2, ainda não há abordagens que possibilitem que tais métricas sejam aplicadas no início do ciclo de vida de desenvolvimento, especificamente na etapa de modelagem de requisitos. Assim, é necessário que modelos ou especificações de requisitos definam elementos que possam ser contabilizados e, eventualmente, sirvam como base para estimativas de esforço e custo mais precisas. Além disso, há poucos trabalhos que deixem explícitos os dados históricos de aplicações Web reais que foram usadas definir métricas apropriadas.

Assim, este capítulo descreve o levantamento de dados conduzido em uma pequena empresa de desenvolvimento de software para a Web, com o propósito de identificar e avaliar os principais indicadores de esforço no desenvolvimento de uma aplicação Web. A partir desse estudo, foram levantados os fatores que mais afetam na complexidade de desenvolvimento de elementos de hipermídia que compõem uma aplicação Web. Esses fatores, por sua vez, foram pontuados e agrupados de forma sistemática em um conjunto de métricas, denominado WHP (*Web Hypermedia Points*). Para dar suporte à aplicação da métrica proposta, estendeu-se o modelo de hipertexto (ou modelo navegacional) da WebML, adicionando-se alguns elementos de hipermídia necessários na forma de unidades de conteúdo.

O capítulo está organizado da seguinte maneira: na Seção 3.2 apresentam-se as unidades de conteúdo propostas para estender o modelo de hipertexto; na Seção 3.3 descreve-se a métrica WHP, em detalhes; na Seção 3.4 apresenta-se uma avaliação inicial da métrica e, finalmente, na Seção 3.5 são delineadas as considerações finais.

### **3.2. Extensão de Elementos da WebML**

Dentre os principais métodos de modelagem conceitual propostos na literatura nos últimos anos, WebML é o mais promissor (Martino, 2007), na medida que permite

especificar aplicações Web complexas fornecendo um conjunto de notações visuais para modelar a estrutura de conteúdo e aspectos navegacionais de uma aplicação. Essa é a principal razão para o sucesso da WebML e motivou a adoção desse método de modelagem e a extensão proposta.

O modelo de hipertexto da WebML foi estendido para modelar visualmente elementos de conteúdo estático e de apresentação, independentemente de tecnologias de implementação. A hipótese inicialmente formulada de que tais elementos desempenham um papel fundamental, principalmente no contexto de aplicações Web de hipermídia, foi confirmada pelo estudo realizado na empresa de desenvolvimento de software para Web, referenciada neste trabalho.

O levantamento de dados tornou, portanto, evidente a necessidade de modelar aspectos estáticos e de apresentação em um nível conceitual adequado, uma vez que esses aspectos possuem atributos específicos que contribuem significativamente para o aumento da complexidade total de desenvolvimento. Além disso, como uma nova motivação para a extensão, verificou-se também que unidades de conteúdo estático e de apresentação permitem satisfazer certos requisitos relacionados a conteúdo e apresentação, derivados de objetivos estratégicos de diferentes *stakeholders*.

No que diz respeito apenas a conteúdo estático, ainda pode-se citar como motivação a importância que esta categoria de conteúdo representa no contexto de aplicações Web mais complexas, favorecendo a simplicidade e estabilidade de páginas, além do tempo curto de resposta a alguma solicitação. Assim, as quatro notações visuais propostas que estendem a WebML incluem *Image Units*, *Multiimage Units*, *Animation Units* e *Text Units*, descritas, em detalhes, a seguir.

### ***Image Units***

Uma *image unit* é usada para publicar qualquer tipo de imagem na aplicação Web, compreendendo, portanto, a representação de gráficos, fotos, mapas, ícones e logomarcas, basicamente. *Image units* são consideradas como unidades de conteúdo de apresentação e podem ser usadas em dois contextos distintos: no contexto estático, quando as imagens são usadas de forma isolada, isto é, não recebem nenhuma informação como parâmetro de entrada, e no contexto de dados, quando existe um *link* de entrada carregando a informação do objeto cuja imagem deve ser exibida.

Percebe-se que no contexto de dados, uma *image unit* pode estar ligada, por meio de um *link* contextual, a uma *data unit*, *multidata unit*, *index unit* ou uma *scroller unit*. Em XML, uma *image unit* é especificada usando-se o elemento IMAGEUNIT e os seguintes atributos: *id*, *entity* (opcional), *width*, *height*, *type* e *background*. O atributo *id* é usado para permitir identificar unicamente uma *image unit*, da mesma forma que na especificação de outras unidades de conteúdo. O atributo *entity*, que é opcional, indica a entidade à qual a *image unit* corresponde. Os atributos *width* e *height* representam, respectivamente, a largura e altura de uma imagem, e os valores contidos nesses atributos podem ser usados como entrada para a geração automática de código.

Finalmente, os atributos *type* e *background* são considerados indicadores do grau de esforço de desenvolvimento requerido pela *image unit*, portanto devem ser especificados. O atributo *type* refere-se ao tipo de imagem (por exemplo, foto, ícone, logomarca, etc.), e o atributo *background* diz respeito ao tipo de fundo de uma imagem. Esses dois atributos e a sua influência exercida na complexidade de desenvolvimento serão descritos, em detalhes, na seção seguinte.

A Figura 17 mostra a notação gráfica para uma *image unit* e o código XML associado. Conforme mostra a Figura, a representação de uma *image unit*, quando usada no contexto estático, também inclui um valor entre os sinais de menor e maior, indicando algum tema ou tópico estático ao qual a imagem se refere.

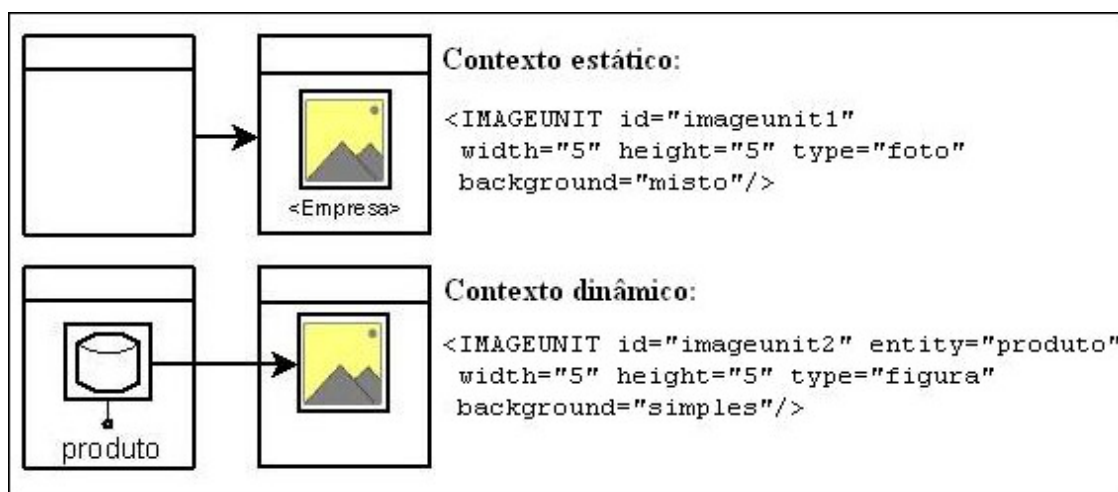


Figura 17: Notação Gráfica para uma *Image Unit* e Código XML Correspondente.



### Multiimage Units

Uma *multiimage unit* é usada para publicar um conjunto de imagens na aplicação Web, repetindo a apresentação de várias *image units*. Assim como no caso de uma *image unit*, uma *multiimage unit* pode ser usada tanto no contexto estático quanto no contexto de dados. A especificação em XML de uma *multiimage unit* possui os atributos *id*, *entity* (opcional) e *imagecount*. Esse último representa a quantidade de imagens a serem publicadas na página. O elemento MULTIIMAGEUNIT deve ser usado para definir uma *multiimage unit*, e deve incluir o elemento IMAGEUNIT aninhado, para cada imagem exibida. A Figura 18 exemplifica a representação gráfica de uma *multiimage unit* e o código XML correspondente.

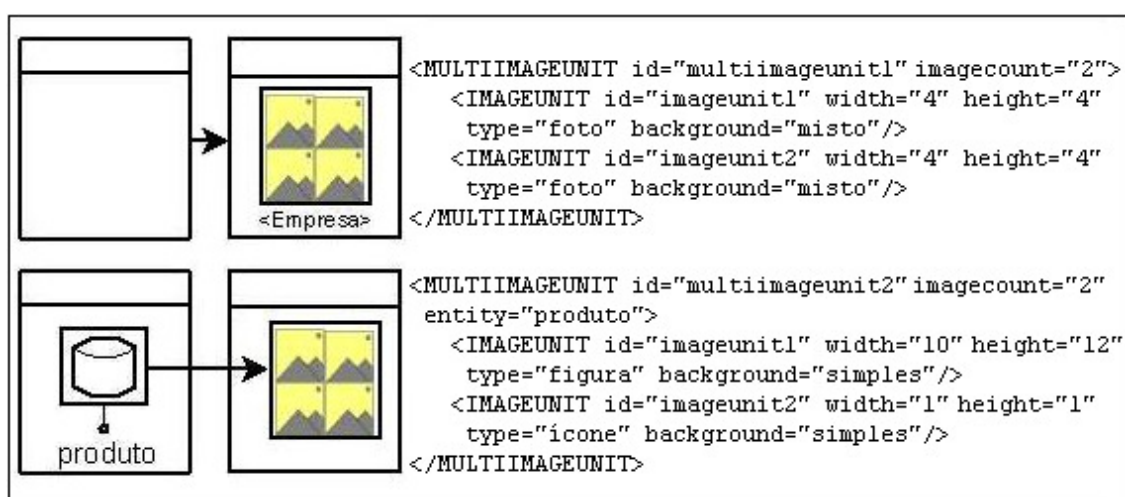


Figura 18: Notação Gráfica para uma *MultiImage Unit* e Código XML Correspondente.

### Animation Units

Uma *animation unit* é usada para publicar uma animação ou qualquer tipo de efeito animado na aplicação Web. *Animation units* também podem ser usadas tanto no contexto estático quanto no contexto de dados, e são especificadas em XML usando-se o elemento ANIMATIONUNIT. Os seguintes atributos devem ser incluídos na especificação: *id*, *entity*, *type* e *imagecount*. O atributo *type* especifica o tipo de animação a ser desenvolvida – *Gif* ou *Flash* –, e o atributo *imagecount* define o número de imagens necessárias para produzir a animação. O grau de influência desempenhado por esses atributos no esforço de desenvolvimento será descrito, em detalhes, na seção seguinte. A Figura 19 ilustra a notação gráfica e o código XML associado.

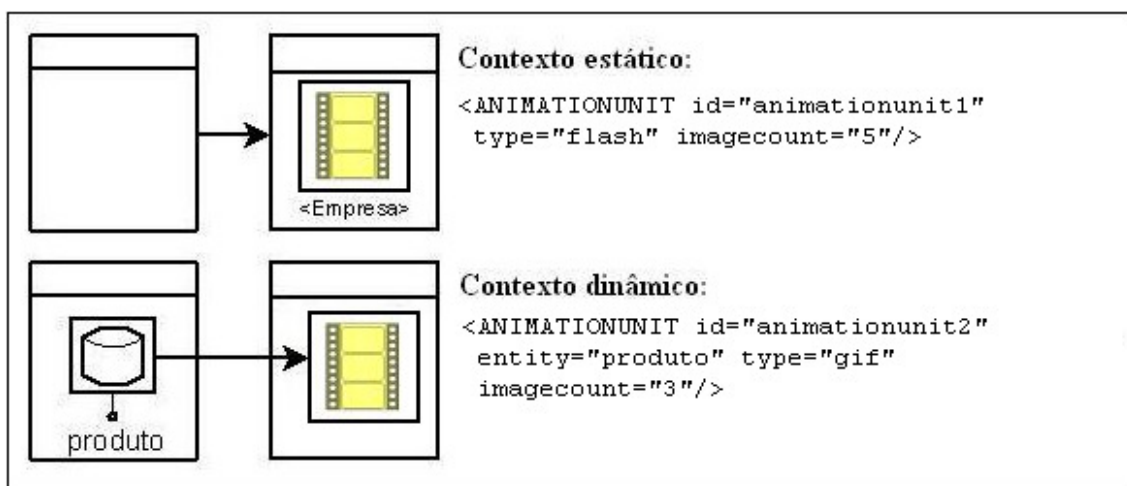


Figura 19: Notação Gráfica para uma *Animation Unit* e Código XML Correspondente.

### Text Units

Uma *text unit* é usada para publicar conteúdo textual estático, representando informações e idéias cruciais que precisam ser exibidas pela aplicação Web, mas que dificilmente se modificarão ao longo do tempo. Sintaticamente, uma *text unit* é representada pelo elemento TEXTUNIT, que deve incluir além do atributo *id*, o atributo *wordcount*, cujo valor indica a quantidade de palavras usadas no texto. A Figura 20 ilustra a notação gráfica e o código XML correspondente.

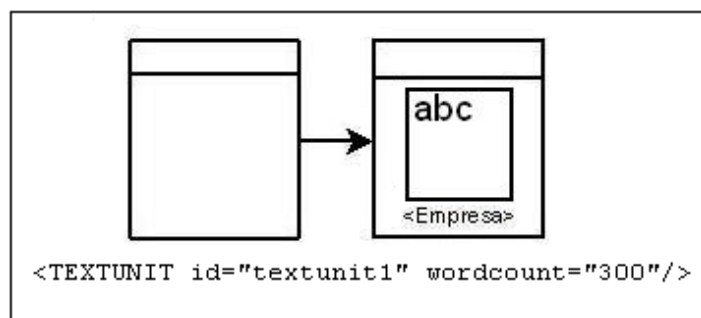
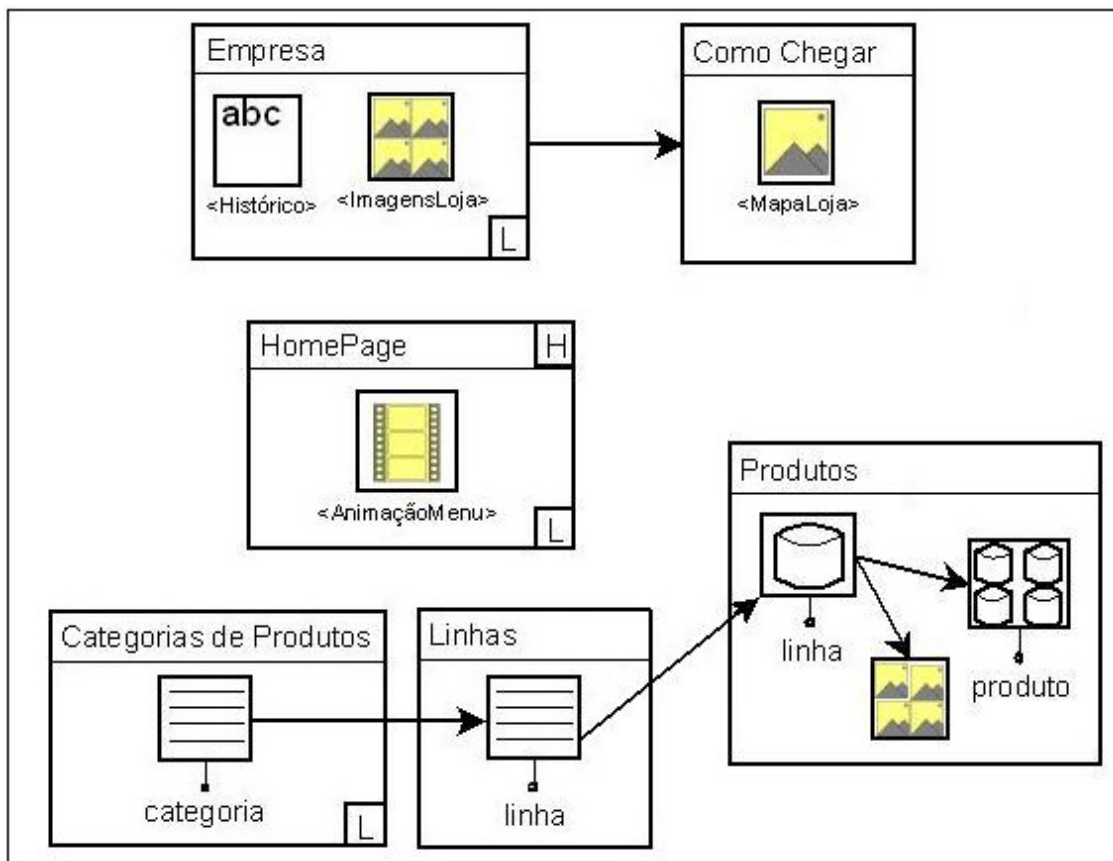


Figura 20: Notação Gráfica para uma *Text Unit* e Código XML Correspondente.

### Exemplo de aplicação

Para ilustrar a utilização do conjunto de notações visuais proposto, é apresentado um trecho do modelo de hipertexto referente à visão pública de uma aplicação Web. Essa aplicação, denominada Tapetes-SC, publica e mantém informações acerca de uma empresa especializada na venda de tapetes, localizada na cidade de São Carlos, SP, Brasil. De acordo com o modelo apresentado na Figura 21, a página “Home” é

composta por apenas uma unidade de conteúdo – uma *animation unit* –, que será computada assim que o usuário acessar a página. A partir da página “Home”, o usuário pode ter acesso às páginas “Empresa” e “Categorias de Produtos”, pois ambas são globalmente acessíveis (*Landmark*).



**Figura 21: Trecho do Modelo de Hipertexto para a aplicação Web Tapetes-SC.**

Na página “Empresa”, o usuário pode ter acesso a um texto falando sobre o histórico da empresa, representado por uma *text unit*, e um conjunto de imagens referentes às construções antigas da loja, representado por uma *multiimage unit*. É importante observar que como a *text unit* é uma unidade de conteúdo estática e a *multiimage unit* está sendo usada no contexto estático, não há *links* contextuais apontando para essas unidades. A página “Como Chegar”, acessada apenas por meio da página “Empresa”, é composta por uma *image unit*, representando um mapa sobre como chegar até a loja.

Na página “Categorias de Produtos”, existe uma *index unit* representando uma lista de categorias de produtos da loja (Tapetes, Carpetes, Produtos Automotivos e

Produtos especiais). O usuário, ao clicar em uma categoria da lista, é levado à outra página contendo uma nova *index unit*, formada pelas linhas de produtos (ou sub-categorias) pertencentes à categoria escolhida na página anterior.

Selecionando uma linha do índice, o usuário terá acesso à página “Produtos”, que exibe informações detalhadas da sub-categoria escolhida, publicadas por uma *data unit*, além de produtos e imagens correspondentes à sub-categoria, publicados, respectivamente, por uma *multidata unit* e uma *multiimage unit*. Nesse caso existe um *link* contextual apontando para uma *multiimage unit* de forma que ela exiba as imagens referentes à linha em questão (contexto de dados). Vale salientar que nenhuma unidade de conteúdo proposta envia informações a outras unidades, independentemente do contexto de uso.

### **3.3. O Processo de Definição da Métrica WHP**

Com base nas métricas identificadas por Mendes et al. (2001), foi realizado o levantamento de dados, conforme já mencionado, a fim de investigar as maiores dificuldades encontradas por uma empresa de desenvolvimento Web em estimar prazos e custos. Assim, o objetivo principal do estudo foi identificar os aspectos que mais influenciam no esforço de desenvolvimento e definir, a partir desses aspectos, métricas de software buscando aumentar a precisão de estimativas, além de melhorar o acompanhamento do projeto e aperfeiçoar a qualidade do software.

O estudo de caso foi realizado na empresa *Linkway*, localizada na cidade de São Carlos, SP, Brasil. Inicialmente, procurou-se identificar os papéis desempenhados pelos membros da equipe de Engenharia da Web na empresa, e suas respectivas atividades. Dessa forma, a pesquisa envolveu desenvolvedores, web designers, editores de conteúdo, especialistas no domínio do negócio, especialistas de suporte e gerentes.

O tempo despendido e a descrição de cada atividade específica de projeto – seja técnica ou não-técnica – que deve ser conduzida durante o desenvolvimento são registrados em planilhas de dados pelos membros da equipe. Assim, os dados de cada planilha foram extraídos e analisados sob a perspectiva dos principais fatores considerados como indicadores de esforço de desenvolvimento. Ao todo, foram analisados registros de 15 aplicações Web concluídas, e as informações contidas nas planilhas sobre cada aplicação foram mapeadas em unidades de conteúdo definidas pela

WebML, e pelas unidades de conteúdo estático e de apresentação, propostas neste trabalho.

As aplicações Web usadas no levantamento de dados possuem atributos diversos e algumas características em comum, apresentadas em diferentes graus de intensidade em cada caso. Foram analisados os dados de aplicações Web de cada um dos paradigmas – paradigma de hipermídia e de transação –, além de algumas aplicações consideradas mais complexas, derivadas essencialmente da combinação de ambos os paradigmas. Para cada elemento de hipermídia (unidade de conteúdo) constituindo cada uma das aplicações Web, foram identificadas propriedades específicas que mais afetam a complexidade de desenvolvimento do elemento de hipermídia em questão. Além de propriedades, em alguns casos também foram identificadas atividades que podem contribuir para o aumento de complexidade.

Os registros de tempos nas planilhas foram agrupados em intervalos, de acordo com cada propriedade ou atividade específica. O limite inferior e o limite superior de cada intervalo foram associados, respectivamente, ao menor e maior valor de tempo registrado para a propriedade ou atividade em questão. A partir disso, foi possível abstrair três grandes intervalos (baixo, médio e alto) e atribuir um valor de pontos de complexidade para cada um desses intervalos, de forma a refletir a diferença de pontos médios (valores médios de tempos para cada intervalo) entre eles.

Uma vez que valores de pontos foram atribuídos aos intervalos de tempos para cada elemento de hipermídia, esses valores foram comparados entre diferentes elementos para se obter a complexidade relativa de cada um deles. Assim, foi possível ajustar os valores de pontos atribuídos a cada elemento, um em relação ao outro. Algumas reuniões realizadas com os membros da equipe de Engenharia da Web da empresa deram apoio à obtenção de pontos de complexidade, e entrevistas individuais com dois web designers permitiram refinar os valores obtidos.

Os resultados do levantamento de dados indicam que aspectos de apresentação, incluindo imagens (representadas por *image units* ou *multiimage units*) e animações (representadas por *animation units*), aliados à complexidade estrutural (relacionamento entre páginas e *links*), são os fatores que, em média, mais influenciam na complexidade

de desenvolvimento de aplicações Web. Os valores de complexidade obtidos para imagens são apresentados na Tabela 3.

**Tabela 3: Pontos de Complexidade para Imagens.**

<b>Complexidade de Imagem (<i>Image Unit</i>)</b>			
	Edição (E)	Otimização (O)	E + O
Fundo Uniforme	Baixa	Baixa	Média
Fundo Misto	Baixa	Média	Alta
Baixa – 2 pontos; Média – 6 pontos; Alta – 12 pontos			

De acordo com a Tabela 3, pode-se verificar que a complexidade de uma imagem é afetada basicamente pelo tempo despendido em duas atividades principais: edição e otimização. A primeira atividade agrupa tarefas simples como ajuste de brilho, contraste e cor, redimensionamento e corte de imagens; já a segunda atividade está relacionada ao processo de limpeza de fundo ou sobreposição de imagens. No caso da atividade de otimização, observa-se que a sua complexidade varia conforme o tipo de fundo da imagem considerada – fundo uniforme ou fundo misto (quando há tons diferentes em uma mesma imagem). Fotos, consideradas como uma categoria especial de imagens, devem ser avaliadas quanto a sua complexidade usando-se os mesmos critérios aplicados às imagens de fundo misto.

**Tabela 4: Pontos de Complexidade para Vetorização de Imagens.**

<b>Complexidade de Vetorização de Imagem (<i>Image Unit</i>)</b>		
Total de Elementos		
1-10	11-20	>20
Baixa	Média	Alta
Baixa – 20 pontos; Média – 45 pontos; Alta – 90 pontos		

Algumas imagens ainda devem ser submetidas a um processo chamado de vetorização. O processo de vetorização consiste na conversão de imagens (formada por *pixels*) para o domínio geométrico. Dessa forma, imagens são armazenadas em forma de arquivos geométricos, de tamanho geralmente menor, e, como resultado, a maioria das operações de processamento é feita de forma mais eficiente. É importante observar que uma imagem, ao ser vetorizada, pode ser redimensionada sem perda de qualidade na resolução. As logomarcas, muito utilizadas principalmente em aplicações Web de e-

*business*, são normalmente submetidas a esse processo. A complexidade envolvida nessa atividade é mostrada na Tabela 4.

Observa-se, pela Tabela 4, que a complexidade de vetorização de uma imagem é afetada essencialmente pela quantidade de elementos (detalhes) a serem vetorizados em uma imagem. Em muitos casos, inicialmente as imagens devem ser digitalizadas antes de serem submetidas ao processo de tratamento. Vale destacar que o tempo consumido durante a atividade de digitalização depende basicamente da quantidade de cores da imagem a ser digitalizada e do equipamento utilizado para digitalização, assim, o esforço gasto nessa atividade deve ser contabilizado separadamente.

Para avaliar a complexidade envolvida no desenvolvimento de animações, é necessário identificar o tipo de animação requerido (*Flash* ou *Gif*), o número de imagens que irão compor a seqüência de efeitos e a quantidade de eventos na página que devem ser capturados (este último relacionado apenas a animações em *Flash*). A Tabela 5 mostra os intervalos de complexidade para uma animação.

**Tabela 5: Pontos de Complexidade para Animações.**

<b>Complexidade de Animação (<i>Animation Unit</i>)</b>				
	Eventos Capturados	Total de Imagens		
		0-3	4-7	>7
Tipo Gif	-	Baixa	Baixa	Média
Tipo Flash	0-10	Baixa	Média	Alta
	+10	Média	Alta	Alta
Baixa – 15 pontos; Média – 60 pontos; Alta – 130 pontos				

Observa-se que o tempo consumido na criação e no tratamento de qualquer animação aumenta conforme aumenta o número de imagens utilizadas para produzir o efeito desejado. Animações em *Gif* são geralmente mais simples que animações em *Flash*, entretanto um efeito criado nesta última tecnologia pode usar menos imagens (quadros) em relação ao mesmo tipo de efeito criado em *Gif*. Essa situação ocorre porque animações em *Flash* podem ser criadas de duas formas distintas: animação por quadros-chave e animação por interpolação.

Animação por quadros-chave envolve a criação de quadros principais da seqüência, e os quadros intermediários são gerados automaticamente. Já a animação por

interpolação exige apenas a criação de dois quadros-chave, e os quadros intermediários restantes são gerados a partir de cálculos de transformações geométricas. É importante ressaltar que os valores de complexidade considerados na Tabela 5 não consideram a complexidade de tratamento de imagens usadas na animação. Essa complexidade deve ser avaliada seguindo-se os mesmos critérios aplicados às imagens, de acordo com as Tabelas 3 e 4.

O número de eventos que devem ser capturados também exerce influência no esforço de desenvolvimento, apenas no caso de animações criadas em *Flash*, conforme é mostrado na Tabela 5.

Já a complexidade de conteúdo textual varia de acordo com a quantidade de palavras usadas no texto, da mesma forma como foi proposto por Cleary (2000). Os valores de complexidade, apresentados na Tabela 6, incluem as atividades de análise e revisão de conteúdo, semelhantes, em muitos pontos, à revisão de um documento escrito. Assim, o conteúdo estático deve ser examinado tanto no nível sintático quanto no semântico. No nível sintático, ortografia, pontuação e gramática devem ser avaliados nos documentos, e no nível semântico, são verificadas a correção, a consistência ao longo de todo o conteúdo textual e a ausência de ambigüidades. Deve-se ressaltar que o tempo gasto na atividade de digitalização ou digitação de textos não é considerado pela métrica.

**Tabela 6: Pontos de Complexidade para Textos.**

<b>Complexidade de Texto (<i>Text Unit</i>)</b>		
Total de Palavras		
0-300	301-800	>800
Baixa	Média	Alta
Baixa – 1 ponto; Média – 2 pontos; Alta – 4 pontos		

Para o caso de *data units* (ou *multidata units*) e *entry units*, a complexidade de desenvolvimento está relacionada à quantidade de atributos – no caso de *data units* – e à quantidade de campos exibidos – no caso de *entry units*, conforme mostra a Tabela 7. Neste ponto, é fundamental ressaltar que os pontos de complexidade definidos na métrica WHP consideram apenas o esforço despendido na criação e análise de atributos ou campos, isto é, os aspectos estáticos de desenvolvimento. A complexidade total



dessas unidades de conteúdo deve ser avaliada aplicando-se o modelo de métrica definido no método Web-SEMP, conforme será explicado no capítulo seguinte.

**Tabela 7: Pontos de Complexidade para Entradas e Saídas de Dados.**

<b>Complexidade de Entrada/Saída de Dados (Data Unit – Entry Unit)</b>		
Total de Atributos/Campos		
1-10	11-20	>20
Baixa	Média	Alta
Baixa – 30 pontos; Média – 50 pontos; Alta – 80 pontos		

No que diz respeito à complexidade estrutural (estrutura de páginas e *links*) de uma aplicação Web, foi possível identificar, através da análise de dados, que existe uma relação entre a densidade de conectividade (número de links dividido pelo número de páginas) e o número de *links* que contribui consideravelmente para o aumento da complexidade total de desenvolvimento de uma aplicação Web. A Tabela 8 mostra o nível de influência exercido por essa relação na complexidade estrutural de aplicações Web.

**Tabela 8: Pontos de Complexidade para Relação Estrutural.**

<b>Complexidade de Relação Estrutural (Page e Link)</b>			
	Total de <i>Links</i>		
Densidade	1-20	20-40	>40
1,0-1,3	Baixa	Baixa	Média
1,3-1,7	Média	Média	Alta
+1,7	Média	Alta	Alta
Baixa – 100 pontos; Média – 300 pontos; Alta – 550 pontos			

Percebe-se, pela Tabela 8, que a complexidade estrutural aumenta conforme aumentam o valor de densidade e a quantidade de *links* na aplicação. Os valores de complexidade apresentados na Tabela 8 compreendem o esforço de interligação entre páginas e objetos de conteúdo, além do esforço envolvido na atividade de teste de navegabilidade. Todas as ligações, incluindo *links* entre páginas (*links* não-contextuais) e *links* entre objetos de conteúdo (*links* contextuais), são testadas para garantir que toda

a sintaxe e semântica de navegação sejam exercitadas a fim de descobrir quaisquer erros, tais como *links* inativos, impróprios ou errados.

Deve-se ressaltar que a duração de atividades de pesquisa ou obtenção de conteúdo não é contabilizada pela métrica WHP. O tipo e a duração de tais atividades, comumente conduzidas durante o processo de desenvolvimento Web, devem ser previstos na Etapa 1 do método Web-SEMP, conforme será explicado no Capítulo seguinte.

### **3.4. Avaliação da Métrica WHP**

Para avaliar a efetividade da métrica WHP assim como efetuar alguns pequenos ajustes nos pontos de complexidade atribuídos aos elementos de hipermídia, a métrica proposta foi aplicada a um subconjunto daquelas aplicações Web que foram usadas no levantamento de dados. Como os dados históricos das aplicações Web já haviam sido coletados sob a perspectiva das unidades de conteúdo definidas na WebML em conjunto com as unidades de conteúdo propostas neste trabalho, foi necessário apenas atribuir pontos de complexidade aos elementos de hipermídia já identificados no estudo.

A aplicação da métrica foi realizada juntamente com a equipe de Engenharia da Web da empresa *Linkway*. Assim, os pontos de complexidade definidos na métrica foram associados a cada elemento de hipermídia exatamente conforme as propriedades específicas apresentadas por cada um. Especificamente, os web designers da equipe auxiliaram na identificação das características de todas as imagens e animações que foram usadas nas aplicações Web estudadas.

Dentre as 15 aplicações Web analisadas no levantamento de dados, foram usadas 10 aplicações apresentando características distintas. A Tabela 9 mostra a quantidade de elementos de hipermídia que compõem cada aplicação. Pode-se observar que do total, sete aplicações são completamente estáticas (letra “E” na coluna Tipo) e três apresentam conteúdo dinâmico derivado dos dados mantidos como parte de um sistema de banco de dados (letra “D” na coluna Tipo). É necessário explicitar que no caso das aplicações Web dinâmicas, os dados correspondem apenas aos aspectos estáticos que compõem a visão pública das aplicações.

**Tabela 9: Dados de Aplicações Web usadas na Avaliação.**

Aplicação Web	<i>Page</i>	<i>Link</i>	<i>Data Unit</i>	<i>Entry Unit</i>	<i>Text Unit</i>	<i>Image Unit</i>	<i>Animation Unit</i>	Tipo
Reiseg	18	28	0	1	3	120	1	E
O2Personal	75	89	0	1	19	100	3	E
Sorts	14	18	0	4	1	11	1	E
Arckpack	42	76	0	1	29	62	1	E
Exatur	95	109	0	1	4	86	1	E
ZPTratores	16	27	0	1	10	228	0	E
Raperfurados	12	12	0	1	4	83	1	E
CMI	247	640	3	4	201	37	1	D
Tapetes	24	49	5	4	5	554	1	D
Landart	10	21	1	4	2	317	1	D

A Tabela 10 mostra o valor total de complexidade atribuído a cada uma dessas aplicações assim como o tempo realmente consumido durante o seu desenvolvimento. Com o auxílio dos web *designers* da equipe, foi possível abstrair apenas o tempo gasto em atividades relacionadas a elementos de hipermídia que compõem uma aplicação Web, visto que as planilhas de dados também continham anotações de tempo sobre atividades de análise e coleta de conteúdo. Informações mais detalhadas sobre a aplicação da métrica em cada aplicação Web podem ser encontradas no Apêndice A.

**Tabela 10: Pontos de Complexidade Atribuídos a cada Aplicação Web.**

Aplicação Web	Pontos	Esforço Real (Horas)
ZPTratores	486	17:33
Raperfurados	526	19:16
Sorts	573	20:14
Reiseg	588	20:22
O2Personal	650	23:01
Arckpack	698	24:50
Exatur	741	26:35
CMI	992	34:25
Landart	1442	53:30
Tapetes	2334	91:20

Observando-se a diferença de horas consumidas e a correspondente diferença de pontos de complexidade para cada par de aplicações Web, constata-se que os valores de

pontos atribuídos são proporcionais à quantidade de horas efetivamente gastas no desenvolvimento. Dessa forma, procurou-se analisar formalmente o grau de associação (correlação) entre o esforço real (representado pela variável dependente  $y$ ) e o número de pontos de complexidade atribuído a cada aplicação Web (representado pela variável independente  $x$ ). Para esse propósito, foi usado o coeficiente de correlação de *Pearson*, de acordo com a seguinte fórmula:

$$r = \frac{n\sum xy - \sum x\sum y}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

O valor de  $n$  na fórmula é a quantidade de aplicações Web usadas no estudo, portanto, nesse caso,  $n = 10$ . O valor de  $r$  está entre -1 e +1 e indica o coeficiente de correlação entre os dois valores, assim quanto maior for o valor de  $r$ , maior será o grau de correlação. Aplicando-se a fórmula aos sites Web, obteve-se um valor de  $r$  igual a 0,99, indicando uma forte correlação positiva entre valores de pontos de complexidade e tempos consumidos no desenvolvimento.

Para determinar uma função entre os valores de pontos de complexidade (variável  $x$  independente) e horas consumidas (variável  $y$  dependente), efetuou-se uma análise de regressão linear, de acordo com a função  $y = a + bx$ . Os valores de  $a$  e  $b$  foram calculados através do Método dos Mínimos Quadrados, usando-se as seguintes fórmulas:

$$a = \frac{\sum y}{n} - b \cdot \frac{\sum x}{n} \quad (1) \quad \text{e} \quad b = \frac{n\sum xy - \sum x\sum y}{n\sum x^2 - (\sum x)^2} \quad (2)$$

Com a aplicação das fórmulas (1) e (2), os valores obtidos foram, respectivamente:  $a = -2,834$  e  $b = 0,0398$ . Portanto, a função entre pontos de complexidade e horas gastas pode ser dada por:

$$y = -2,834 + 0,0398x \Rightarrow \text{esforço} = -2,834 + 0,0398 \cdot (\text{pontos})$$

Uma vez que se obteve a função linear representando a relação entre pontos de complexidade e horas gastas, foi possível aplicar a fórmula a cada site Web a fim de comparar o tempo estimado pela técnica de regressão com o tempo real gasto (em horas

decimais). A comparação foi efetuada calculando-se, para cada aplicação, o valor de MRE (*Magnitude of Relative Error*), dado pela seguinte fórmula (em porcentagem):

$$MRE = \left| \frac{Esforço_{Real} - Esforço_{Estimado}}{Esforço_{Real}} \right| * 100$$

De acordo com Conte et al. (1986), o valor limite aceitável do MRE deve ser igual ou menor a 25%, isto é, o erro gerado pela aplicação de uma técnica de estimativa de esforço deve ser igual ou inferior a 25% do esforço real. O MRE pode ser visto como um valor numérico capaz de avaliar o poder de previsão de modelos de estimativa, assim um valor de MRE baixo dá indícios de um bom modelo de previsão.

A Tabela 11 mostra o valor de esforço gerado pela fórmula para cada aplicação Web, e o valor real de esforço, além do MRE correspondente. De acordo com a Tabela, observa-se que o maior valor de MRE é 6,5% para o site CMI, seguido pelos sites Raperfurados (6,0%) e ZPTratores (5,9%). No caso do site CMI, conforme é mostrado no Apêndice A, a sua densidade de conectividade é muito alta (2,59) pois trata-se de uma aplicação usada essencialmente como um repositório de documentos acessíveis on-line. Esse fator propiciou a geração de um valor um pouco acima do tempo real consumido.

**Tabela 11: Cálculo de Esforço e MRE para cada Projeto Web.**

Aplicação Web	Pontos	Esforço Real (Horas decimais)	Esforço Estimado (Horas decimais)	MRE
Reiseg	588	20,36	20,56	0,9%
O2Personal	650	23,00	23,03	0,1%
Sorts	573	20,23	19,97	1,2%
Arckpack	698	24,83	24,94	0,4%
Exatur	741	26,58	26,65	0,2%
ZPTratores	486	17,55	16,51	5,9%
Raperfurados	526	19,26	18,10	6,0%
CMI	992	34,41	36,64	6,5%
Tapetes	2334	91,33	90,06	1,4%
Landart	1442	53,50	54,55	1,9%

Como sete de todas as aplicações Web usadas na avaliação são puramente estáticas, foi possível comparar os valores obtidos através da fórmula com a técnica

Web Points, discutida, em detalhes, no Capítulo 4. A Tabela 12 mostra a comparação de ambas as técnicas e o valor de MRE apresentado pela aplicação da técnica Web Points. Percebe-se, pela Tabela, que a técnica Web Points apresentou valores muito acima dos valores reais de esforço, e o valor do MRE apenas foi utilizado, neste caso, para formalizar a discrepância entre os valores estimados e os valores reais.

**Tabela 12: Cálculo de Esforço em Web Points e Valor de MRE para Sites Web Estáticos.**

Site	Complexidade de Página			Web Points	Esforço Estimado	Esforço Real	MRE (%)
	Baixa	Média	Alta				
Reiseg	14	3	1	81	$81 * 0,5 = 40,5$	20,36	98,9
O2Personal	26	1	0	110	$110 * 0,5 = 55,5$	23,00	141
Sorts	14	0	0	56	$56 * 0,5 = 28$	20,23	38,4
Arckpack	41	1	0	170	$170 * 0,5 = 85$	24,83	242,3
Exatur	95	0	0	380	$380 * 0,5 = 190$	26,58	614,8
ZPTratores	6	10	0	84	$84 * 0,5 = 42$	17,55	139,31
Raperfurados	11	1	0	60	$60 * 0,5 = 30$	19,26	55,7

A diferença de esforço observada pela técnica Web Points está relacionada ao tipo de julgamento de complexidade proposto pela técnica. Web Points avalia quão complexo é um site Web que possui muitas páginas; entretanto, conforme pode ser observado pelas características das aplicações usadas no estudo, pode haver sites Web com muitas páginas formadas por pouco conteúdo, ou ainda pode haver uma quantidade consideravelmente maior de páginas baseadas em texto do que páginas baseadas em conteúdo multimídia.

De acordo com os modelos de previsão gerados a partir das métricas propostas por Mendes et al. (2001), percebe-se que o esforço de desenvolvimento é dado basicamente pela quantidade de elementos de hipermídia que compõem uma aplicação Web e por algumas métricas de complexidade estrutural (conectividade e densidade de conectividade).

Especificamente, o modelo gerado para avaliar o esforço de extensão (*length*) é calculado por meio das seguintes variáveis: “quantidade de páginas”, “espaço ocupado por uma página (em *bytes*)” e “tamanho total de código reusado”. Desconsiderando-se as duas últimas variáveis, as quais são difíceis de serem medidas na fase de modelagem de análise, a quantidade de páginas de uma aplicação Web continua sendo o fator-chave

para o modelo de estimativa. Já o modelo gerado para avaliar o esforço de complexidade (*complexity*) é baseado na variável “conectividade”, isto é, o esforço de complexidade estrutural aumenta apenas conforme aumenta a quantidade de *links* na aplicação Web. A Tabela 13 mostra as duas fórmulas obtidas por Mendes et al. (2001).

**Tabela 13: Modelos de Estimativa Gerados por Mendes et al. (2001).**

Fórmula relacionada à extensão	$-3,158 + 0,558 * \text{Número de Páginas} + 0,03937 * \text{Espaço Alocado} + 0,02795 * \text{Tamanho de Código Reusado}$
Fórmula relacionada à complexidade	$20,911 + 0,0280 * \text{Conectividade}$

Apesar dos resultados apresentados pela aplicação da métrica WHP serem promissores, deve-se considerar cuidadosamente a sua aplicação em outros contextos, visto que a métrica foi derivada de dados históricos de uma única empresa. Por um lado, isso possibilitou resultados mais apropriados durante a avaliação mas, por outro, os valores de MRE apresentados na Tabela 12 podem não ser mantidos, caso a métrica seja aplicada em outros contextos industriais.

Ressalta-se também que a função obtida entre pontos de complexidade e horas gastas pode ser usada em contextos nos quais ainda não há informações precisas sobre as habilidades de cada desenvolvedor ou *web designer*. Nos casos em que certas características da equipe de Engenharia da Web já são conhecidas, recomenda-se o uso do NDE de cada membro para a conversão de pontos em horas, conforme comentado no Capítulo 2.

### **3.5. Considerações Finais**

Neste capítulo, apresentou-se a métrica WHP – *Web Hypermedia Points*, proposta para avaliar a complexidade de elementos de hipermídia que compõem uma aplicação Web. Esses elementos de hipermídia incluem as unidades de conteúdos básicas definidas pela WebML e as unidades de conteúdo estático e de apresentação propostas neste trabalho. Dessa forma, a aplicação da métrica deve ser apoiada pelo modelo de hipertexto de uma aplicação Web, o qual é constituído pelos elementos a serem contabilizados.

A métrica WHP foi elaborada a partir de um levantamento de dados realizado em uma pequena companhia de desenvolvimento de software para Web. Com base em

dados históricos, foi possível identificar fatores que afetam na complexidade de desenvolvimento total e atribuir pontos de complexidade a cada elemento de hipermídia, de acordo com tais fatores.

Uma avaliação inicial da efetividade da métrica WHP também foi apresentada com o intuito de verificar se os pontos atribuídos para cada elemento de hipermídia estavam refletindo, em termos proporcionais, as horas consumidas para desenvolvê-lo. Embora os resultados tenham sido positivos em relação à efetividade da métrica, é importante ressaltar que, como a métrica foi aplicada a um subconjunto dos sites cujos dados foram usados no estudo, os mesmos resultados podem não se repetir, caso a métrica seja aplicada em outros contextos.



# CAPÍTULO 4

## O MÉTODO WEB-SEMP

---

### **4.1. Considerações Iniciais**

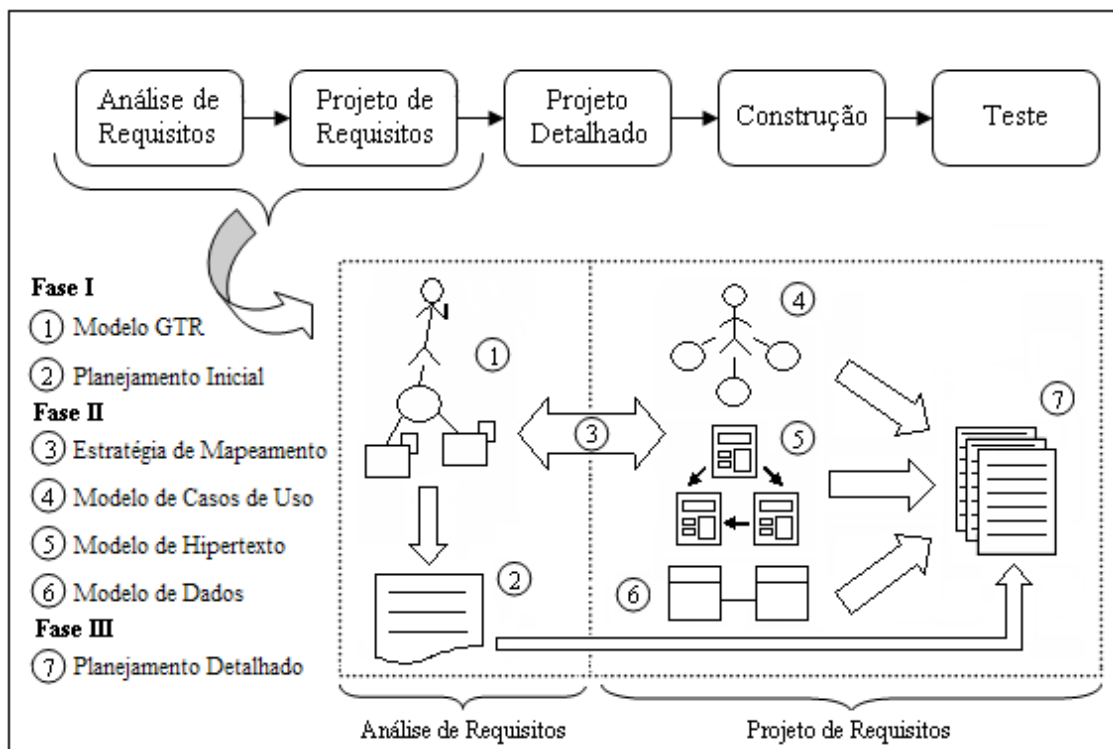
Um modelo conceitual de projeto para aplicações Web deve ser expressivo, de forma que permita representar características cruciais dessas aplicações, mas, ao mesmo tempo, deve ser fácil de ser entendido, exigindo. Embora haja alguns métodos de elicitação de requisitos e de projeto propostos na literatura, não existem diretrizes que conduzam, de forma sistemática, a transição entre uma especificação inicial de requisitos e um modelo de projeto conceitual.

Assim, com o objetivo de contribuir nas atividades de elicitação, modelagem e planejamento de aplicações Web, este capítulo apresenta o método Web-SEMP – *Web System Elicitation, Modeling and Planning*. O método, estruturado de forma a dar apoio efetivo ao projetista na elaboração de um modelo conceitual, é composto por três fases. São elas: Fase I – Elicitação de Requisitos e Planejamento Inicial, Fase II – Aplicação da Estratégia de Mapeamento e Fase III – Aplicação do Modelo de Métrica. Cada uma dessas fases fornece diretrizes para a elaboração de aspectos ortogonais de um modelo conceitual de projeto.

O capítulo está organizado da seguinte forma: na Seção 4.2 apresenta-se o método Web-SEMP em linhas gerais; nas Seções 4.3, 4.4 e 4.5 apresentam-se, passo a passo, as Fases I, II e III, respectivamente, e, finalmente, na Seção 4.6 apresentam-se as Considerações Finais.

### **4.2. Uma Visão Geral do Método Web-SEMP**

O método Web-SEMP dá apoio às atividades de elicitação, modelagem e planejamento de aplicações Web. No contexto do processo de desenvolvimento apresentado no Capítulo 2, o método compreende as fases de Análise de Requisitos e Projeto de Requisitos, conforme mostra a Figura 22.



**Figura 22: O método Web-SEMP no contexto do Processo de Desenvolvimento Web.**

A Fase I – Elicitação de Requisitos e Planejamento Inicial corresponde às atividades de concepção, levantamento e especificação de requisitos. Para essas atividades, o método Web-SEMP adota uma abordagem orientada a objetivos, denominada AWARE (1), explicada, em detalhes, no Capítulo 2. Dessa forma, o processo de ER definido pelo método Web-SEMP é baseado em uma abordagem orientada a *stakeholder*, levando em consideração não apenas as necessidades dos usuários finais da aplicação Web, mas também os objetivos estratégicos e de comunicação de todos os *stakeholders* relevantes.

Durante a Fase I, através da atividade de Planejamento Inicial (2), ainda é possível prever informalmente certos tipos de tarefas não-técnicas, como por exemplo, busca de conteúdos, que deverão ser conduzidas em fases subsequentes do desenvolvimento. De acordo com a Figura 22, a atividade de Planejamento Inicial deve ser baseada nos diagramas GTR construídos, e a saída de tal planejamento pode servir de entrada para a atividade de Planejamento Detalhado definida pelo método Web-SEMP (7).

A Fase II – Aplicação da Estratégia de Mapeamento (3) é baseada na premissa de que uma discussão com *stakeholders* em torno de modelos de requisitos tende a ser

ineficiente ou enganosa, devido à natureza abstrata de objetivos e requisitos. Pela experiência obtida no Estudo de Caso, descrito no Capítulo 5, utilizar especificações iniciais de requisitos nas discussões com *stakeholders* pode dificultar o entendimento sobre a importância relativa de seus objetivos e, portanto, impedir a obtenção de um *feedback* mais concreto.

Com base nesse contexto, a estratégia de mapeamento tem por objetivo auxiliar na construção de modelos conceituais de projeto, focando em tópicos específicos a partir de uma especificação inicial de requisitos, conforme mostra a Figura 22. Assim, a estratégia de mapeamento define um conjunto de diretrizes para derivar idéias de projeto de forma a permitir uma discussão mais concreta com *stakeholders* e auxiliar nas atividades de validação e elicitação de novos requisitos. A Fase II, dividida em etapas, permite obter os seguintes modelos: (4) modelos de casos de uso funcionais e navegacionais; (5) modelos iniciais de estrutura de hipertexto usando as notações definidas pela WebML e as que foram propostas neste trabalho; e (6) entidades e relacionamentos mais relevantes representados no modelo de dados correspondendo à aplicação inteira.

A Fase III – Aplicação do Modelo de Métrica (7) baseia-se nas três categorias de modelos de projeto de requisitos para contabilizar as funcionalidades e os elementos de hipermídia que constituem a aplicação Web. Dessa forma, considera-se que os três modelos fornecem informações de entrada suficientes para a aplicação do modelo de métrica durante a atividade de Planejamento Detalhado, conforme é representado pela Figura 22.

Pode-se considerar que as duas primeiras fases definidas pelo método Web-SEMP correspondem essencialmente às atividades de levantamento e modelagem de requisitos, enquanto que a terceira fase está relacionada à atividade de planejamento, com base em aspectos de projeto obtidos nas fases anteriores. A seguir, cada fase será discutida em detalhes.

### **4.3. Fase I – Elicitação de Requisitos e Planejamento Inicial**

A Fase I do método Web-SEMP é composta por duas etapas: Etapa de Elicitação de Requisitos e Etapa de Planejamento Inicial. Portanto, nessa fase, o engenheiro de requisitos deve entender o escopo e a natureza do problema a ser resolvido e especificar

as necessidades e objetivos de todos os *stakeholders*. Baseando-se nas primeiras reuniões e nas especificações iniciais de requisitos, pode-se prever ainda a necessidade de condução de algumas atividades não-técnicas. Nas próximas sub-seções serão descritas as duas etapas que compõem a Fase I – Elicitação de Requisitos e Planejamento Inicial.

### 4.3.1. Etapa 1: Elicitação de Requisitos

**Propósito:** Identificar todos os *stakeholders* envolvidos no projeto e especificar seus respectivos objetivos com relação à aplicação a ser desenvolvida.

**Entrada:** Essa atividade não possui nenhum artefato de entrada.

**Saída:** Diagrama GTR (*Goal-Task-Requirement*) proposto pelo modelo AWARE para cada *stakeholder* e Formulário de Atividades Não-Técnicas.

#### Passo 1

Nesse passo, engenheiros de requisitos devem identificar cada uma das visões da aplicação a ser desenvolvida, de acordo com as categorias de público-alvo. Dessa forma, é possível determinar a necessidade de modelar uma visão pública e visões privadas, cujo conteúdo é específico para cada classe de usuários finais. No caso de aplicações Web mais complexas, é necessário também identificar as visões de gerenciamento de conteúdos específicos. Inicialmente, o engenheiro de requisitos deve criar uma lista de todos os *stakeholders*, isto é, todos aqueles que possuem algum interesse ou objetivo que precisa ser satisfeito pela aplicação.

Como existem muitos *stakeholders* diferentes, os objetivos e, conseqüentemente, os requisitos devem ser explorados a partir de muitos pontos de vista diferentes. Por exemplo, o grupo de marketing está interessado em funções e características que excitam o mercado em potencial, tornando o novo sistema fácil de vender. Os gerentes de negócio estão interessados em um conjunto de características que possam ser construídas dentro do orçamento e que estejam prontas a satisfazer janelas de mercado bem definidas. Os usuários finais podem querer características que lhes sejam familiares e fáceis de aprender e de usar. Finalmente, engenheiros de suporte e manutenção podem focar a manutenibilidade do software. Assim, cada um desses *stakeholders* contribuirá com informações para o processo de engenharia de requisitos.

## Passo 2

Para cada visão da aplicação identificada, deve-se repetir os seguintes sub-passos:

### Passo 2.1

O propósito desse sub-passo é identificar um subconjunto de *stakeholders* da lista criada no Passo 1, que possuem objetivos em relação à visão identificada. Dessa forma, cada visão pode estar relacionada a um ou vários *stakeholders* (incluindo a classe de usuários finais que terão acesso a essa visão).

### Passo 2.2

Para cada *stakeholder* identificado, deve-se elaborar o diagrama GTR proposto pelo modelo AWARE. Portanto, engenheiros de requisitos devem definir, de forma hierárquica, um conjunto de objetivos que podem ser decompostos em sub-objetivos e tarefas, e, posteriormente, transformados em requisitos de hipermídia – incluindo tanto requisitos de hipermídia funcionais quanto não-funcionais.

Durante a construção dos diagramas GTR, é importante, muitas vezes, atribuir prioridades a objetivos, *stakeholders* e requisitos, conforme proposto pelo modelo AWARE. A priorização é fundamental, caso haja muitos *stakeholders* e objetivos envolvidos no projeto, pois assim podem-se tratar interesses conflitantes e tomar decisões importantes acerca de objetivos e requisitos a serem satisfeitos pela aplicação.

A Figura 23 mostra um trecho do diagrama GTR referente a uma aplicação Web de um fornecedor de tapetes. É importante observar na Figura 23 a legenda que mostra a correspondência entre cada sigla que consta nos requisitos do diagrama GTR e o seu significado, de acordo com a taxonomia de requisitos proposta pelo modelo AWARE. Essa legenda vale para todas as figuras que apresentarem diagramas GTR.

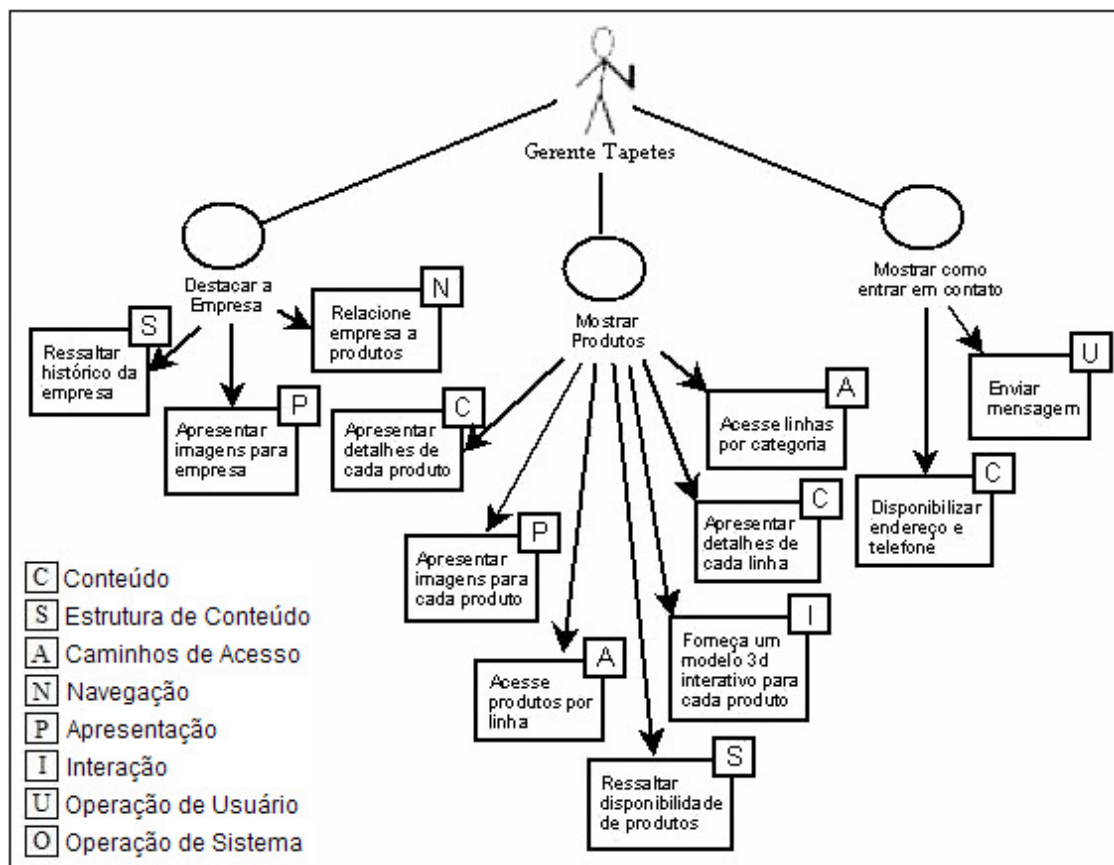


Figura 23: Diagrama GTR para o Gerente de um Fornecedor de Tapetes.

#### 4.3.2. Etapa 2: Planejamento Inicial

**Propósito:** Prever o tipo e a duração de atividades não-técnicas de projeto, com base nas reuniões realizadas com todos os *stakeholders* e com base nos diagramas GTR iniciais.

**Entrada:** Diagramas GTR.

**Saída:** Formulário com a descrição e a duração estimada para cada atividade não-técnica necessária durante o projeto.

O propósito dessa etapa é prever a quantidade, o tipo e a duração de atividades não-técnicas que deverão ser conduzidas concomitantemente a outras atividades de modelagem de análise e de projeto. Essas atividades são comumente conduzidas no desenvolvimento Web em virtude de todo o espectro de conteúdo a ser fornecido pela aplicação. É importante observar que a previsão, nesse caso, é inteiramente baseada na experiência de membros de uma equipe de Engenharia da Web.

Assim, com base, essencialmente, nos requisitos de conteúdo, apresentação e interação definidos nos diagramas GTR, é possível prever atividades de busca e coleta de diversos tipos de conteúdo (textos, imagens, figuras, fotografias, logomarcas, etc.), assim como atividades de análise para revisar e organizar as informações coletadas. As reuniões com os *stakeholders* podem auxiliar na previsão de tais atividades, uma vez que se pode calcular, aproximadamente, a carga de conteúdo a ser pesquisado de acordo com a carga total necessária e a carga de conteúdo fornecida pelos *stakeholders*. Um exemplo de um formulário preenchido com atividades não-técnicas pode ser observado na Tabela 14.

**Tabela 14: Exemplo de Atividades Não-Técnicas Previstas.**

Descrição	Duração estimada
Pesquisa de 5 imagens de tapetes e carpetes	40 minutos
Análise de 2 arquivos em pdf contendo informações sobre a empresa	1 hora
Organização de material enviado pelo cliente para disponibilizar na página como <i>link</i> de download	50 minutos

De acordo com essas atividades, é possível adaptar o fluxo de trabalho durante as fases subseqüentes no processo de desenvolvimento. O resultado da execução de tais atividades ainda permite revelar certas características de conteúdo multimídia que terão influência na complexidade total de desenvolvimento. A análise de certos tipos de conteúdo que deverão ser disponibilizados para *download* (uma prática comum no desenvolvimento Web) também deve ser prevista nessa etapa.

#### **4.4. Fase II – Aplicação da Estratégia de Mapeamento**

A Fase II do método Web-SEMP é composta por três etapas: Etapa de Identificação de Aspectos de Composição, Etapa de Identificação de Aspectos de Navegação e Etapa de Identificação de Aspectos de Funcionalidade.

Nessa fase, adota-se o princípio fundamental da separação de interesses, de maneira que cada etapa represente uma dimensão diferente da aplicação. A separação de interesses permite que analistas dêem foco em dimensões cruciais de projeto, de acordo com as características da aplicação Web a ser desenvolvida. Esse princípio ainda permite inspecionar os diagramas GTR e avaliar a sua completeza, influenciando, assim, na usabilidade e na qualidade final da aplicação.

É importante ressaltar que cada etapa deve ser repetida para cada uma das visões da aplicação identificadas na fase anterior.

#### 4.4.1. Etapa 1: Identificação de Aspectos de Composição

**Propósito:** Entender os requisitos relacionados ao conteúdo nos diagramas GTR para construir o sub-modelo de composição na WebML estendida.

**Entrada:** Diagrama GTR proposto pelo AWARE para cada *stakeholder*.

**Saída:** Formulário Requisito x Tema x *Stakeholder* (RTS) e sub-modelo de composição na WebML estendida.

##### Passo 1

Nesse passo, o objetivo é numerar todos os requisitos de hipermídia que constam nos diagramas GTR para uma dada visão. Essa numeração é importante para que se possam listar e organizar os requisitos de hipermídia que deram origem ao tema, assim como todos os outros requisitos relacionados. A Figura 24 mostra o diagrama GTR da Figura 23, no qual foi adicionada a referida numeração.

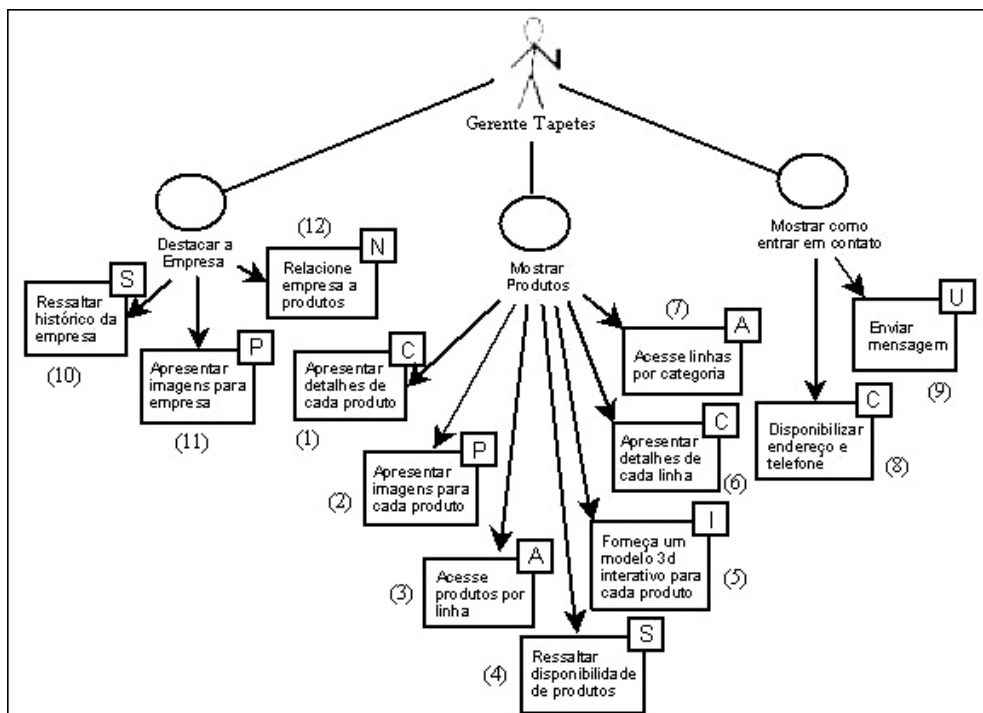


Figura 24: Diagrama GTR Numerado para o Fornecedor de Tapetes.



## Passo 2

Nesse passo, o propósito é identificar todos os candidatos a temas nos diagramas GTR para uma dada visão e preencher o formulário Tema x Requisito x *Stakeholder* (TRS). O formulário TRS possui como objetivo registrar os temas identificados bem como os requisitos de hipermídia relacionados a cada tema. Assim, devem-se analisar substantivos e cláusulas substantivas que constam em requisitos de conteúdo e de estrutura de conteúdo (aqueles marcados com a letra “C” e “S”), e listá-los na coluna “Tema” do formulário TRS. Os números associados aos requisitos referentes a cada um dos temas devem ser listados na coluna “Requisitos Relacionados” do formulário TRS.

A decisão de inclusão de temas no formulário TRS é um tanto subjetiva, e uma avaliação posterior pode causar o descarte ou a inclusão de algum tema não-identificado. Além disso, é necessário estabelecer quais temas cujas informações devem ser mantidas como parte de um sistema de banco de dados (temas dinâmicos), e quais temas cujas idéias e mensagens dificilmente se modificarão ao longo do tempo (temas estáticos). A letra “E” deve ser anotada no formulário TRS para temas estáticos e a letra “D” para temas dinâmicos, os quais vão compor o modelo de dados em forma de entidades. Os *stakeholders* também devem ser registrados no Formulário TRS. A Tabela 15 exemplifica um formulário TRS, preenchido com os temas identificados a partir do diagrama GTR representado pela Figura 23.

**Tabela 15: Exemplo de um Formulário TRS.**

Tema	Requisitos Relacionados	<i>Stakeholder</i>	E/D
Produto	1,2,3,4,5,11,12	Gerente	D
Linha	6,7	Gerente	D
Contato	8,9	Gerente	E
Empresa	10,11,12	Gerente	E

A partir desse ponto, devem-se repetir os passos 3 a 6 para cada tema listado no formulário TRS. Para cada um dos passos (3-6), que corresponde a um domínio de requisito, devem-se verificar requisitos redundantes de um mesmo domínio, isto é, requisitos escritos de formas diferentes, porém sugerindo semanticamente o mesmo tipo de conteúdo. Pode haver redundâncias no diagrama GTR de um *stakeholder* – intra-

*stakeholder* – ou em diferentes diagramas – *inter-stakeholder*. Esses requisitos devem ser eliminados dos diagramas GTR e do formulário TRS.

Além de identificar redundâncias, é necessário verificar requisitos conflitantes associados a *stakeholders* diferentes. Caso haja, o engenheiro de requisitos deverá decidir sobre como esse conflito será resolvido, baseando-se nas prioridades atribuídas aos *stakeholders* envolvidos e nas prioridades que cada *stakeholder* atribuiu ao objetivo que derivou o requisito.

Uma vez que foram tratados todos os temas, deve-se seguir para a Etapa 2 – Identificação de Aspectos de Navegação.

### **Passo 3**

O objetivo desse passo é mapear requisitos de conteúdo referentes a um tema em unidades de dados definidas na WebML estendida. Os seguintes sub-passos devem ser executados:

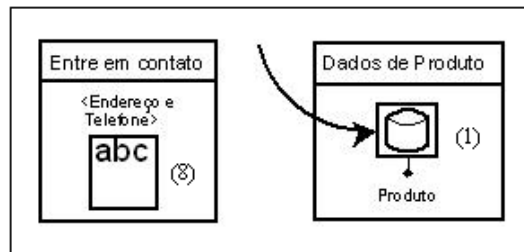
#### **Passo 3.1**

Caso o tema seja estático, requisitos de conteúdo serão mapeados em *text units*. Uma página principal deve ser criada para o tema, e o analista deve decidir se a mensagem ou idéia representada por cada *text unit* deve ser disposta em outras páginas, conectadas à página principal, por meio de um *link* não-contextual. Como exemplo, o requisito de conteúdo referente ao tema “Contato” no diagrama GTR da Figura 22 será mapeado em uma *text unit*, disposta em uma página para apresentar informações ao usuário sobre como entrar em contato com a empresa. A Figura 23 ilustra esse mapeamento.

#### **Passo 3.2**

Caso o tema seja dinâmico, requisitos de conteúdo relacionados a um objeto específico de conteúdo serão mapeados em *data units*. Se o requisito expressar a necessidade de apresentar vários objetos em um mesmo espaço navegacional, isto é, ao mesmo tempo, o requisito de conteúdo será mapeado, então, em uma *multidata unit*. Da mesma forma que no sub-passo anterior, uma página principal deve ser criada para o tema. Como exemplo, o requisito de conteúdo referente ao tema “Produto” no diagrama

GTR da Figura 24 será mapeado em uma *data unit*, disposta em uma página para apresentar dados de um produto específico, conforme mostra a Figura 25. Vale ressaltar que o *link* contextual representado nessa etapa mostra apenas que a *data unit* deve receber a informação sobre qual produto será publicado na página.



**Figura 25: Exemplo de Mapeamento de Requisitos de Conteúdo na WebML Estendida.**

#### Passo 4

O objetivo desse passo é analisar requisitos de estrutura de conteúdo referentes a um tema. A partir desses requisitos, é possível determinar se há a necessidade de evidenciar ou detalhar alguma idéia ou mensagem específica referente a um tema. Caso haja, o projetista deve decidir como essa informação será apresentada na aplicação. Nesse ponto, é fundamental observar que requisitos de estrutura de conteúdo podem ser reduzidos em requisitos de apresentação ou interação e, portanto, podem ser mapeados seguindo-se as mesmas regras aplicadas a esses dois domínios de requisitos.

No diagrama GTR da Figura 24, o requisito “ressaltar disponibilidade de produtos” poderia ser satisfeito usando-se algum recurso de apresentação, como uma proporção de fonte maior, um tom de cor diferente, etc. Nesse caso, como será mostrado no Passo 5, esse tipo de requisito deve servir de entrada para a fase de Projeto Detalhado, no processo de desenvolvimento.

Considerando-se ainda o mesmo diagrama GTR, o requisito “ressaltar histórico da empresa” poderia ser resolvido por meio da publicação de detalhes sobre eventos passados relacionados à empresa (usando uma *text unit*, por exemplo). Fotos antigas também poderiam ser publicadas buscando aumentar o interesse do usuário pela informação apresentada. A partir dos requisitos de estrutura de conteúdo, novas idéias ou mensagens acerca de um tema podem surgir e devem ser mapeados seguindo-se as mesmas regras aplicadas aos requisitos de conteúdo, conforme o Passo 4.

## Passo 5

Nesse passo, o objetivo é identificar todos os requisitos de apresentação relacionados a um tema. Os seguintes sub-passos devem ser executados:

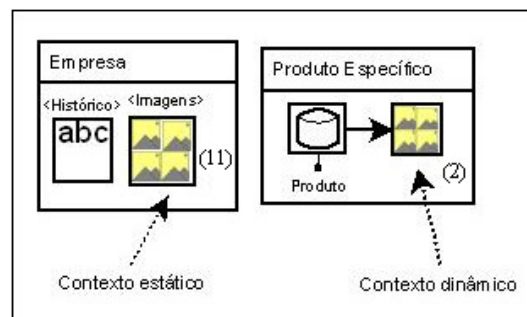
### Passo 5.1

Caso o tema seja estático, requisitos de apresentação relacionados a representações gráficas (figuras, fotos, logomarcas, etc.) podem ser mapeados em *image units* – ou *multiimage units*. Já os requisitos de apresentação relacionados a efeitos animados podem ser mapeados em *animation units*. Em qualquer um dos casos, essas três unidades de apresentação devem ser dispostas na mesma página (contexto estático) na qual é apresentada a mensagem ou idéia referenciada pelo requisito de apresentação.

No diagrama GTR da Figura 24, o requisito “apresentar imagens para a empresa” deve ser mapeado em uma *multiimage unit*, disposta na mesma página na qual são publicadas informações sobre a empresa ou sobre seu histórico. A Figura 26 ilustra esse mapeamento.

### Passo 5.2

Caso o tema seja dinâmico, as regras de mapeamento, no que diz respeito às unidades de apresentação usadas, são as mesmas. Nesse caso, essas unidades devem estar ligadas a outras unidades de conteúdo a fim de receber informação do objeto cuja imagem ou animação será exibida (contexto de dados). A Figura 26 mostra como o requisito “apresentar imagens para cada produto” foi mapeado.



**Figura 26: Exemplo de Mapeamento de Requisitos de Apresentação na WebML Estendida.**

### Passo 5.3

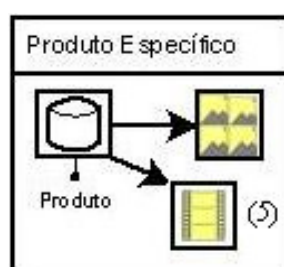
Requisitos de apresentação podem estar relacionados a conteúdos específicos sobre um tema, disponibilizados na forma de arquivos externos para *download*. *Links* para *download* não são representados no modelo de hipertexto, mas a quantidade de *links* assim como o conteúdo relacionado devem ter sido previstos na Etapa de Planejamento Inicial da Fase I do método Web-SEMP.

### Passo 5.4

Requisitos de apresentação podem fornecer indicações sobre o posicionamento físico (leiaute) de determinados elementos de conteúdo ou sobre os estilos de fonte e cores a serem usados em determinadas partes do conteúdo ou em determinadas áreas da aplicação Web. Em todos esses casos, esses requisitos servem de entrada para a fase de Projeto Detalhado no ciclo de vida de desenvolvimento.

### Passo 6

Nesse passo, o objetivo é identificar todos os requisitos de interação relacionados a um tema. Requisitos de interação podem expressar a necessidade de criação de animações mais complexas que permitem a interação do usuário, ou ainda podem representar certos requisitos de operação de usuário que exigem recursos ou habilidades específicas.



**Figura 27: Exemplo de Mapeamento de Requisitos de Interação na WebML Estendida.**

Como exemplo, no diagrama GTR da Figura 24, o requisito “forneça um modelo 3d interativo para cada produto” poderia ser mapeado em uma *animation unit*, ligada à *data unit* responsável pela publicação dos dados de um produto, conforme mostra a Figura 27.

#### 4.4.2. Etapa 2: Identificação de Aspectos de Navegação

**Propósito:** Entender os requisitos de navegação nos diagramas GTR e identificar relacionamentos para interconectar unidades de conteúdo e páginas no modelo de hipertexto da WebML estendida.

**Entrada:** Diagrama GTR proposto pelo AWARE para cada *stakeholder*, formulário TRS e sub-modelo de composição da WebML estendida.

**Saída:** Modelo de hipertexto formado tanto por aspectos de composição quanto por aspectos de navegação e modelo de casos de uso navegacionais.

##### Passo 1

Nesse passo, o objetivo é identificar todos os requisitos de navegação nos diagramas GTR. É importante observar que temas podem ser relacionados por meio de *links* contextuais ou não-contextuais, seguindo-se as regras de validade do modelo de hipertexto, definidas em [Ceri et al., 2000]. Para cada *link*, o projetista deve verificar como o espaço navegacional, formado por elementos de conteúdo referentes ao tema, será modificado, isto é, quais transformações navegacionais irão ocorrer na medida que o usuário atravessar o *link*, seja contextual ou não-contextual.

Os seguintes sub-passos devem ser executados:

##### Passo 1.1

Caso ambos os temas sejam estáticos, o requisito de navegação será mapeado em um *link* não-contextual entre as páginas que contêm as partes de conteúdo semanticamente relacionadas. O projetista deve decidir qual será o sentido de navegação entre os temas.

##### Passo 1.2

Caso ambos os temas sejam dinâmicos, o requisito de navegação será mapeado em um *link* contextual entre as unidades de conteúdo responsáveis pela publicação de dados dos objetos referenciados no requisito. Nesse caso, o projetista deve observar o sentido de navegação entre os temas e, com base nessa observação, verificar a necessidade de utilizar uma *index unit* para possibilitar essa navegação. Requisitos de

navegação relacionados a temas dinâmicos devem servir como entrada para o modelo de dados da aplicação Web, possibilitando a derivação de relacionamentos entre entidades.

### **Passo 1.3**

Caso um dos temas seja estático e o outro seja dinâmico, deverá haver, em qualquer sentido, um *link* não-contextual entre as páginas correspondentes aos temas. O projetista deve decidir apenas qual será o sentido de navegação. Como exemplo, no diagrama GTR da Figura 22, o requisito “relacione empresa a produtos” deve ser mapeado em um *link* não-contextual entre as páginas principais dos temas.

### **Passo 2**

Nesse passo, o objetivo é definir como cada tema será acessado. Os temas principais normalmente são apresentados em páginas globalmente acessíveis (páginas do tipo *landmark*), uma vez que são os tópicos de maior relevância na aplicação Web. Assim, cada visão deve conter uma página *Home*, a partir da qual será possível acessar os principais tópicos da aplicação Web. Os seguintes sub-passos devem ser executados:

#### **Passo 2.1**

Caso o requisito expresse a necessidade de apresentar dados sobre objetos de conteúdo referentes a um tema de uma maneira guiada, *scroller units* serão usadas em conjunto com a unidade de conteúdo publicando dados sobre tais objetos.

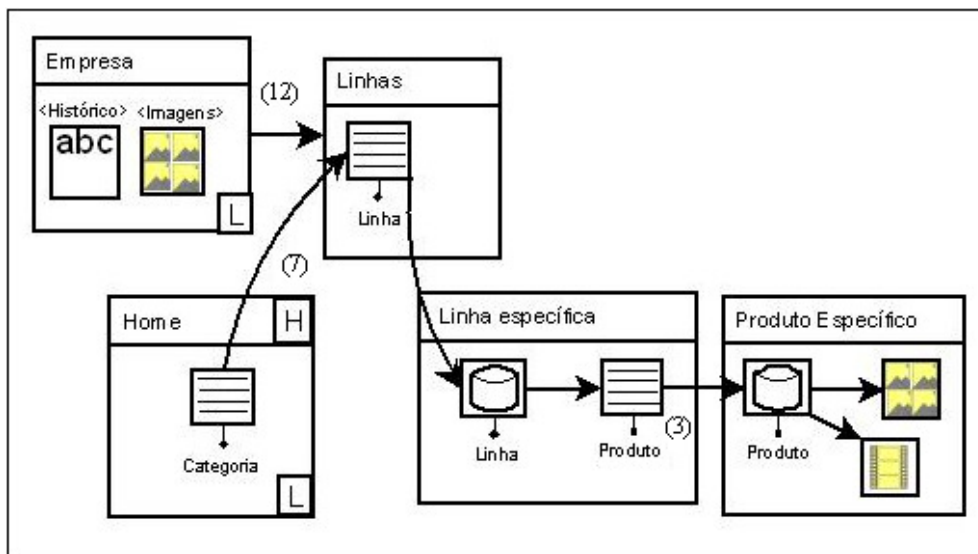
#### **Passo 2.2**

Caso o requisito expresse a possibilidade de acessar objetos de conteúdo referentes a um tema por meio de uma lista – índice –, *index units* serão usadas em conjunto com a unidade de conteúdo publicando dados sobre tais objetos. Nesse caso, os elementos do índice devem ser atributos de temas relacionados ou ainda podem representar temas de acesso, isto é, temas criados apenas para facilitar o acesso aos principais temas da aplicação Web. Como um padrão de navegação, *index units* podem ser adicionadas automaticamente no modelo de hipertexto para permitir acessar um conjunto de objetos referentes a um tema.

### Passo 2.3

Formulários de busca também podem ser usados para permitir acessar objetos de conteúdo de um tema. Tais formulários podem ser usados em conjunto com *index units* ou em conjunto com outras unidades de conteúdo tais como *data units* ou *multidata units*.

A Figura 28 mostra o resultado da aplicação dos Passos 1 e 2 para o diagrama GTR da Figura 24.



**Figura 28: Exemplo de Aplicação da Etapa 2 para o diagrama GTR do Fornecedor de Tapetes.**

### Passo 3

Nesse passo, o objetivo é identificar um conjunto de casos de uso navegacionais. A aplicação Web pode ser descrita por meio de um conjunto de casos de uso navegacionais.

#### Passo 3.1

Casos de uso navegacionais estáticos são formados essencialmente por um conjunto de páginas e *links* não-contextuais entre essas páginas. Assim, pode-se reconhecer um caso de uso navegacional estático por meio de temas estáticos que compõem a aplicação Web, assim como relacionamentos navegacionais entre esses



temas. Um exemplo de caso de uso navegacional estático para o diagrama GTR da Figura 22 seria “acessar empresa”.

### **Passo 3.2**

Casos de uso navegacionais dinâmicos são formados por um conjunto de páginas, *links* entre páginas e *links* contextuais entre unidades de conteúdo que constituem cada uma das páginas. Cada cenário de um caso de uso navegacional desse tipo é formado basicamente por uma seqüência [estrutura de acesso – *link* contextual] diferente. Um exemplo de caso de uso navegacional dinâmico para o diagrama GTR da Figura 22 seria “acessar produtos”.

### **4.4.3. Etapa 3: Identificação de Aspectos de Funcionalidade**

**Propósito:** Entender os requisitos de operação de usuário e de sistema nos diagramas GTR para identificar casos de uso funcionais.

**Entrada:** Diagrama GTR proposto pelo AWARE para cada *stakeholder*.

**Saída:** Modelo de casos de uso navegacionais e funcionais.

#### **Passo 1**

Nesse passo, o objetivo é identificar todos os requisitos de operação de usuário relacionados a uma dada visão. Um requisito de operação de usuário pode ser mapeado em um caso de uso funcional ou em um cenário de um caso de uso funcional. Nesse último caso, a tarefa que permitiu derivar o requisito de operação será mapeada no caso de uso funcional. A partir da análise de requisitos de operação de usuário é possível também identificar associações de inclusão, extensão ou generalização entre casos de uso. Para o diagrama GTR da Figura 24, pode ser identificado um caso de uso funcional, a partir do requisito “enviar mensagem” referente ao tema “Contato”.

#### **Passo 2**

O propósito desse passo é identificar todos os requisitos de operação de sistema. Esses requisitos correspondem a operações que devem ser executadas pelo sistema como parte de um ou mais casos de uso funcionais.

### **4.5. Fase III – Aplicação do Modelo de Métrica**

**Propósito:** Avaliar a complexidade total de desenvolvimento da aplicação Web aplicando a métrica WHP – *Web Hypermedia Points* em conjunto com a técnica Pontos por Casos de Uso.

**Entrada:** Modelo de hipertexto na WebML estendida para cada visão da aplicação, modelo de dados e modelos de casos de uso funcionais e navegacionais.

**Saída:** Tabelas contendo o valor total de Pontos de Hiperímia (Métrica WHP) e valor de Pontos por Casos de Uso.

#### **4.5.1. Etapa 1: Contagem de Caso de Uso Navegacional Estático**

Casos de uso navegacionais estáticos são aqueles que se referem aos temas estáticos de uma visão da aplicação Web. Esses temas são organizados por meio de um conjunto de páginas independentes e *links* não-contextuais entre tais páginas.

Dessa forma, a métrica WHP deve ser aplicada, avaliando-se a complexidade de elementos de hiperímia de cada página. Tais elementos incluem *text units*, *image units*, *multiimage units* e *animation units*. A complexidade de cada elemento de hiperímia deve ser verificada de acordo com as tabelas de pontos de complexidade, propostas pela métrica WHP, apresentada no Capítulo 3.

Após a avaliação de complexidade de todos os elementos de hiperímia constituindo uma página, cada *link* saindo da página deve ser contado, seja um *link* interno – apontando para outras páginas da mesma aplicação Web –, um *link* externo – apontando para outras aplicações –, ou um *link* para *download* de arquivos. *Links* para *download* não são representados no modelo de hipertexto, entretanto a quantidade de tais *links* que devem ser dispostos em cada página deve ter sido prevista na Etapa de Planejamento Inicial da Fase I, do método Web-SEMP.

A quantidade de páginas e *links* de cada caso de uso navegacional estático deve servir como entrada para avaliar a complexidade de relação estrutural da aplicação Web, conforme é proposto pela métrica WHP.

### 4.5.2. Etapa 2: Contagem de Caso de Uso Navegacional Dinâmico

Casos de uso navegacionais dinâmicos são aqueles que se referem aos temas dinâmicos de uma visão da aplicação Web. Para casos de uso navegacionais dinâmicos, deve-se avaliar tanto a complexidade de hipermídia quanto a complexidade funcional.

Da mesma forma que na etapa anterior, a métrica WHP deve ser aplicada, avaliando-se a complexidade de elementos de hipermídia de cada página. Cada *link* saindo da página deve ser contado, seja um *link* contextual ou não-contextual. Nesse ponto, é fundamental observar que *links* de transporte (aqueles em que não há intervenção do usuário) e *links* gerados dinamicamente (como em um índice, por exemplo) não devem ser contabilizados.

Para avaliar a complexidade de hipermídia relacionada a uma *data unit* ou *multidata unit*, a quantidade de atributos de cada entidade, cujas informações são publicadas por uma dessas unidades de conteúdo, serve como entrada para a aplicação da métrica WHP. Para uma *entry unit*, a quantidade de campos pode ser contabilizada de acordo com a entidade cujos dados devem ser inseridos pelo usuário no formulário, fornecendo subsídios para a aplicação da métrica WHP.

O caso de uso ainda deve ser avaliado quanto à sua complexidade funcional, aplicando-se a técnica Pontos por Casos de Uso. A técnica deve ser usada normalmente, por meio da quantidade de cenários ou por meio da quantidade de temas dinâmicos de um caso de uso navegacional. Cada cenário, conforme já mencionado neste trabalho, corresponde a um caminho navegacional diferente, o qual pode ser percorrido pelo usuário seguindo estruturas de acesso e *links* contextuais inerentes a essas estruturas.

### 4.5.3. Etapa 3: Contagem de Caso de Uso Funcional

Os casos de uso funcionais devem ser avaliados quanto à sua complexidade aplicando-se apenas a técnica Pontos por Casos de Uso. É importante observar que qualquer ligação gerada automaticamente como parte do processamento de um caso de uso funcional não deve ser contabilizada usando-se a métrica WHP.

#### 4.5.4. Etapa 4: Avaliação de Complexidade Total da Aplicação Web

Nessa etapa, os valores de pontos de complexidade atribuídos a cada caso de uso navegacional (estático e dinâmico) devem ser somados e colocados em uma tabela de pontos de complexidade de hipermídia. Além disso, o somatório de *links* e de páginas para cada caso de uso deve ser usado como entrada para avaliar a complexidade estrutural de uma visão da aplicação Web. O valor de pontos de complexidade deve ser convertido em horas usando-se o NDE de cada membro da equipe de desenvolvimento, incluindo tanto desenvolvedores quanto *web designers*.

Caso não haja dados históricos detalhados sobre projetos anteriores ou ainda sobre habilidades específicas da equipe de desenvolvimento, o valor total de pontos de hipermídia pode ser convertido em horas através da fórmula derivada para avaliar a complexidade de hipermídia, proposta neste trabalho. Para a complexidade funcional, recomenda-se, nesse caso, usar a quantidade de homens-hora proposta por Karner (1993).

#### 4.6. Considerações Finais

Neste capítulo apresentou-se o método Web-SEMP para dar apoio às atividades de elicitação, modelagem e planejamento para aplicações Web. O método é composto por três fases: Fase I – Elicitação de Requisitos e Planejamento Inicial, a qual adota uma abordagem orientada a objetivos para levantar e especificar requisitos de hipermídia, e com base essencialmente nesses requisitos, permite ainda planejar certas tarefas não-técnicas necessárias para o desenvolvimento; Fase II – Aplicação da Estratégia de Mapeamento, que permite derivar aspectos cruciais de projeto, focando, em particular, no modelo de hipertexto da aplicação Web; e Fase III – Aplicação do Modelo de Métrica, permitindo contabilizar tanto aspectos funcionais quanto de hipermídia de uma aplicação Web com base nos modelos construídos na Fase II.

No próximo capítulo apresenta-se um estudo de caso que foi conduzido em uma pequena empresa de desenvolvimento de software para Web para uma primeira avaliação do método Web-SEMP, proposto neste trabalho.

# CAPÍTULO 5

## EXEMPLO DE APLICAÇÃO NO SISTEMA REAL DA MIREX

---

### ***5.1. Considerações Iniciais***

O método Web-SEMP, apresentado no capítulo anterior, foi aplicado em um estudo de caso referente ao desenvolvimento de uma aplicação Web real, o qual foi realizado na empresa Linkway, localizada na cidade de São Carlos – SP, Brasil. O objetivo desse capítulo é relatar o estudo de caso conduzido para colocar em prática os principais conceitos utilizados pelo método Web-SEMP com o intuito de fazer uma primeira avaliação de sua efetividade.

O capítulo está organizado da seguinte maneira: na Seção 5.2 apresenta-se uma breve descrição da aplicação Web usada no estudo de caso; na Seção 5.3 descreve-se a aplicação da Fase I do método; na Seção 5.4 apresenta-se a aplicação de cada passo constituindo a Fase II; na Seção 5.5 é apresentada a Fase III do método Web-SEMP e, finalmente, na Seção 5.6 apresentam-se as considerações finais.

### ***5.2. Caracterização do Estudo de Caso***

A Mirex é uma empresa nacional voltada ao desenvolvimento de tecnologias para o controle de formigas cortadeiras. Em busca de maiores benefícios para o seu negócio, a empresa solicitou o desenvolvimento de uma aplicação Web para disponibilizar informações acerca de produtos e serviços oferecidos. Portanto, para aumentar a competitividade usando-se os principais recursos fornecidos pelo ambiente Web, dois objetivos cruciais de alto-nível foram estabelecidos pela empresa: “atrair novos clientes” e “facilitar contatos”.

Para contribuir com o primeiro objetivo, a aplicação Web deveria servir como uma ferramenta de comunicação efetiva buscando aumentar o interesse de visitantes pelos produtos apresentados. Já para contribuir com o segundo objetivo, a aplicação deveria servir como um recurso fácil de ser entendido e usado, para que clientes atuais e

clientes em potencial pudessem entrar em contato com a empresa. Focando basicamente nesses dois objetivos, as primeiras reuniões entre analistas e *stakeholders* foram conduzidas, e o método Web-SEMP pôde ser aplicado, desde a Fase I até a Fase III, conforme será detalhado nas próximas seções.

É importante ressaltar que a equipe de desenvolvimento da empresa que participou do estudo de caso não apresentava nenhum conhecimento prévio quanto às abordagens adotadas e estendidas pelo método Web-SEMP. Assim, dois *web designers*, um desenvolvedor e o gerente de projetos Web da equipe receberam um treinamento inicial, dividido em duas sessões de uma hora para a utilização do método proposto durante o desenvolvimento da aplicação Web.

### **5.3. Aplicação da Fase I**

Na Fase I, de acordo com o método Web-SEMP, conduziram-se as primeiras reuniões com os principais *stakeholders*, e, dessa forma, além dos dois objetivos anteriormente citados, outros objetivos foram levantados e especificados em diagramas GTR. Nessa fase, também foi constatada a necessidade de conduzir duas atividades não-técnicas durante o desenvolvimento da aplicação Web. A seguir, descreve-se a aplicação das Etapas 1 e 2 compondo a Fase I do Método Web-SEMP.

#### **5.3.1. Etapa 1: Elicitação de Requisitos**

##### **Passo 1:**

De acordo com as primeiras reuniões realizadas entre a empresa *Linkway* e a *Mirex*, foram identificados três *stakeholders* para a aplicação Web: um dos diretores da *Mirex*, um representante do departamento de *marketing* e um representante de balconista. Além disso, durante as reuniões identificou-se um perfil de usuário correspondente à comunidade que representa o público-alvo da empresa. Esse perfil, aqui denominado cliente *Mirex*, modela uma variedade de categorias de usuários finais, abordando tanto clientes em potencial quanto aqueles que já são clientes da empresa. Esse perfil de usuário ainda pode representar visitantes ocasionais ou usuários com alguma experiência prévia com relação ao site.

A partir de discussões iniciais com os principais *stakeholders* identificados, foi possível determinar a necessidade de se modelar uma visão pública e duas visões

privadas, sendo uma voltada apenas para balconistas e a outra voltada para o gerenciamento de conteúdos apresentados tanto na visão pública quanto na visão acessada por balconistas.

## Passo 2:

**2.1.** Para cada visão determinada, identificou-se o subconjunto de *stakeholders* de todos aqueles levantados no Passo 1, cujos objetivos precisavam ser satisfeitos pela visão. O conteúdo apresentado pela visão pública deve satisfazer requisitos relacionados ao diretor da Mirex, ao representante de *marketing* e também ao perfil de usuário, representando o público-alvo da aplicação. Uma das visões privadas está voltada para balconistas, mas também deve satisfazer objetivos do diretor da Mirex. Já a outra é a visão que o administrador geral (representando o diretor) acessará para gerenciar todas as categorias de conteúdo disponibilizadas tanto na visão pública quanto na visão voltada para balconistas. A Tabela 16 mostra as visões identificadas e os *stakeholders* relacionados a cada uma delas.

**Tabela 16: Visões da Aplicação e Stakeholders Correspondentes.**

<b>Visão da Aplicação</b>	<b>Stakeholders</b>
Visão para Clientes (Visão Pública)	Diretor da Mirex Representante de Marketing Cliente Mirex
Visão para Balconistas (Visão Privada)	Representante de Balconista Diretor da Mirex
Visão para Gerenciamento (Visão Privada)	Diretor da Mirex

**2.2.** Para os *stakeholders* de cada visão, elaboraram-se os diagramas GTR. No que diz respeito à **visão pública**, as Figuras 29, 30 e 31 mostram, respectivamente, os diagramas referentes ao diretor da Mirex, ao representante de *marketing* e ao cliente Mirex. De acordo com a Figura 29, os objetivos de mais alto-nível relacionados ao diretor da Mirex (e, conseqüentemente, à empresa como um todo) são: “Contratar Funcionários”, “Falar sobre a Empresa”, “Apresentar Formigas”, “Informar Métodos de Controle”, “Facilitar Contatos” e “Atrair Novos Clientes”.

Dentre esses, conforme já comentado, os dois últimos objetivos foram considerados como cruciais pelo diretor. O objetivo “Facilitar Contatos” foi mapeado

em quatro requisitos de hipermídia para tornar mais fácil a comunicação entre clientes e a empresa. Já o objetivo “Atrair Novos Clientes” foi decomposto no sub-objetivo “Comunicar Fatores Críticos de Sucesso” que, por sua vez, foi decomposto em outros dois sub-objetivos: “Mostrar Produtos” e “Mostrar Tecnologias”.



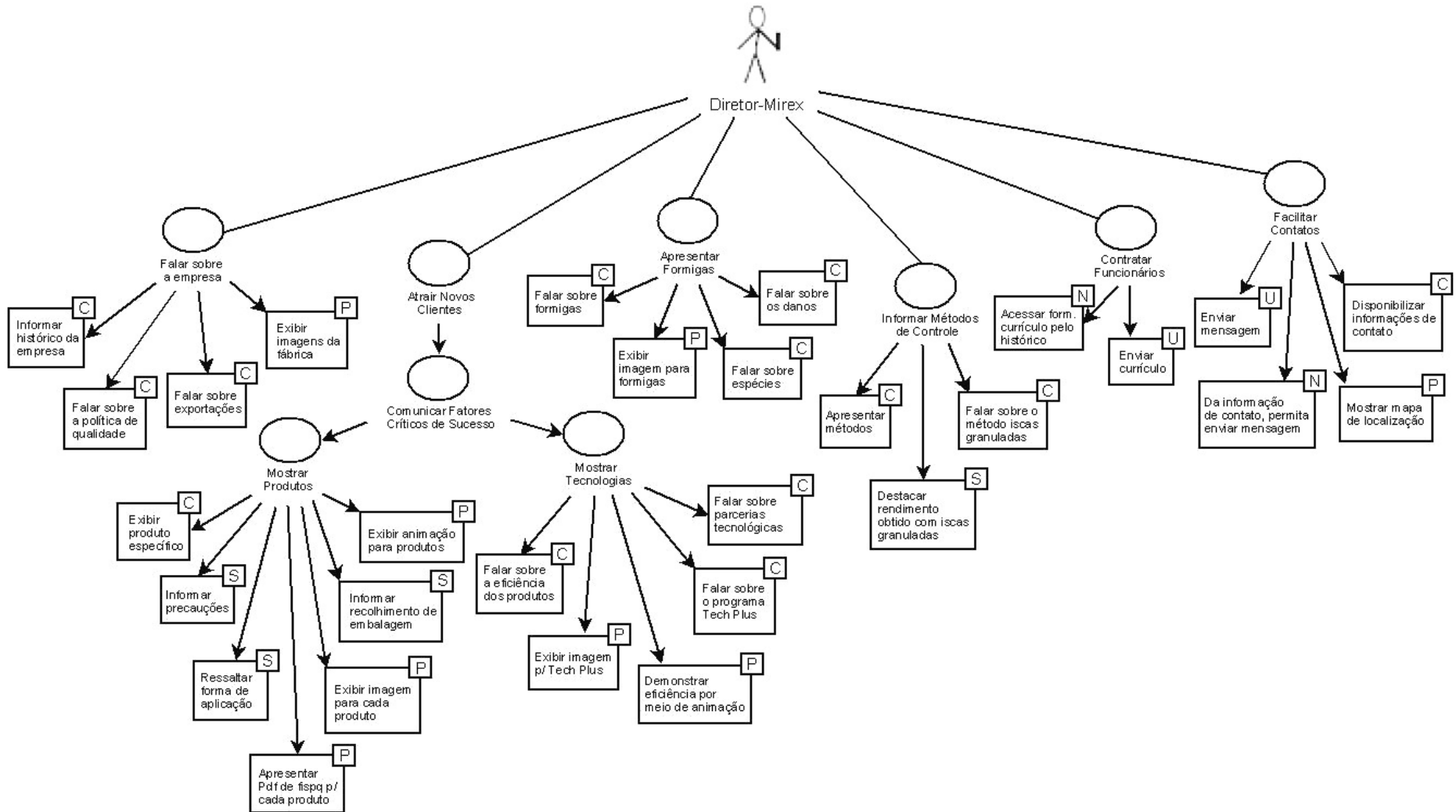


Figura 29: Diagrama GTR para o Diretor da Mirex.

Foi constatado, através das reuniões, que uma estratégia razoável para convencer visitantes do site a se tornarem clientes da Mirex era comunicar os fatores críticos de sucesso, ressaltando as principais características dos produtos fornecidos e as tecnologias usadas no combate às formigas.

O diagrama GTR para o representante de *marketing*, ilustrado na Figura 30, é composto apenas por dois objetivos: “Comunicar Notícias” e “Apresentar Tecnologias”. É importante observar que o departamento de *marketing* possuía outros objetivos, mas esses foram transformados nos mesmos requisitos já representados no diagrama do diretor. Portanto, decidiu-se não representar tais objetivos no diagrama para o representante de *marketing*. Essa observação é importante no sentido de que pode haver objetivos iguais possuídos por mais de um *stakeholder*; entretanto, de acordo com cada ponto de vista analisado, requisitos diferentes podem ser derivados a partir de um mesmo objetivo, compartilhado por vários *stakeholders*.

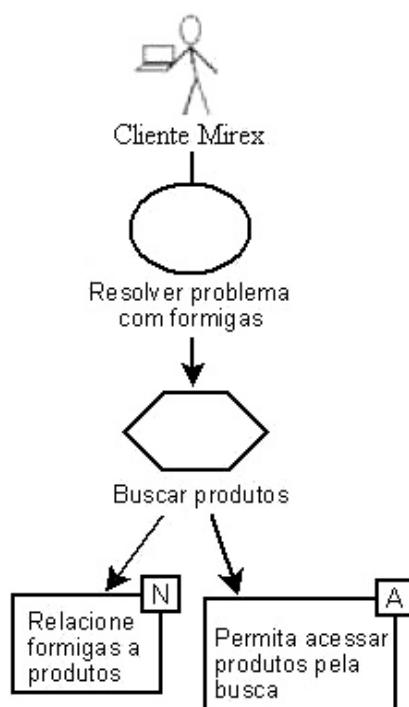
O objetivo “Apresentar Tecnologias”, possuído pelo departamento de *marketing*, está relacionado também ao diretor da Mirex. Esse objetivo foi representado novamente apenas para enfatizar a necessidade de detalhar o programa “TechPlus” – um programa de serviços técnicos específicos da Mirex – e mostrar detalhes sobre o nível de eficiência dos produtos apresentados. Assim, para satisfazer esse objetivo, foram derivados dois novos requisitos de estrutura de conteúdo, conforme pode ser observado na Figura 30. Já o objetivo “Comunicar Notícias” refere-se ao anúncio de eventos e feiras envolvendo a empresa, além de lançamentos de novos produtos.



**Figura 30: Diagrama GTR para o Representante de Marketing.**

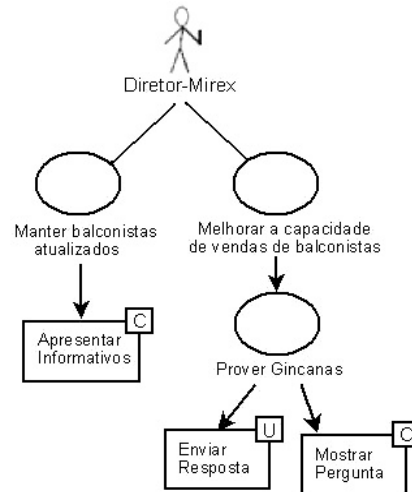
Apesar de boa parte dos requisitos satisfazendo necessidades de um possível cliente da Mirex já estarem especificados nos diagramas GTR anteriores (para o diretor da empresa e para o representante de *marketing*), identificou-se um objetivo do ponto de vista de um visitante do site, possibilitando a derivação de novos requisitos. O objetivo identificado “Resolver Problema com Formigas” foi decomposto em uma tarefa chamada “Buscar Produtos”, a qual possibilitou a derivação de dois novos requisitos. A Figura 31 ilustra o diagrama GTR para o cliente Mirex.

Para o cliente Mirex, é importante observar que o requisito “Permita acessar produtos pela busca” foi derivado para tornar mais fácil a busca de produtos no site, considerando-se, principalmente, visitantes ocasionais. Já o requisito “Relacione formigas a produtos” foi identificado considerando-se que um cliente em potencial poderia querer acessar, inicialmente, informações sobre formigas para, em seguida, conhecer alguns dos produtos fornecidos pela empresa.



**Figura 31: Diagrama GTR para o Cliente Mirex.**

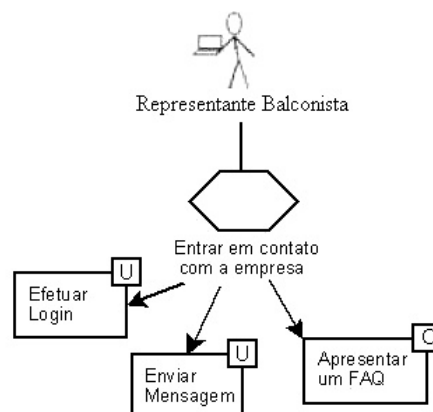
No que diz respeito à **visão voltada para balconistas (visão privada)**, as Figuras 32 e 33 mostram, respectivamente, os diagramas referentes ao diretor da Mirex e ao representante de balconistas.



**Figura 32: Diagrama GTR para o Diretor da Mirex (Visão de Balconistas).**

Conforme pode ser observado na Figura 32, o diretor da Mirex possui como principais objetivos relacionados a essa visão, manter balconistas das lojas atualizados, fornecendo informativos periodicamente, e melhorar a capacidade de vendas de tais balconistas. Para satisfazer esse último objetivo, a estratégia estipulada pelo diretor foi lançar gincanas com perguntas relacionadas a informações mais detalhadas da Mirex e de todos os seus produtos, permitindo, assim, treinar a distância seus funcionários.

Do ponto de vista do representante de balconista, foi identificado um único objetivo: “Entrar em Contato com a Empresa”. Para satisfazer esse objetivo, dois requisitos de operação de usuário foram derivados a fim de fornecer um canal de comunicação simples entre funcionários e empresa, conforme mostra a Figura 33.



**Figura 33: Diagrama GTR para o Representante de Balconista.**

Para a **visão privada de gerenciamento de conteúdos**, elaborou-se o diagrama GTR formado por tarefas e requisitos de hipermídia dando suporte a essas tarefas, de acordo com a Figura 34. Percebe-se que a grande maioria dos requisitos derivados é da categoria de operação de usuário, uma situação inversa em relação à visão pública, na qual os requisitos de conteúdo são preponderantes. É importante ressaltar que valores de prioridades não foram explorados nesse estudo de caso devido ao número relativamente baixo de *stakeholders* e objetivos.

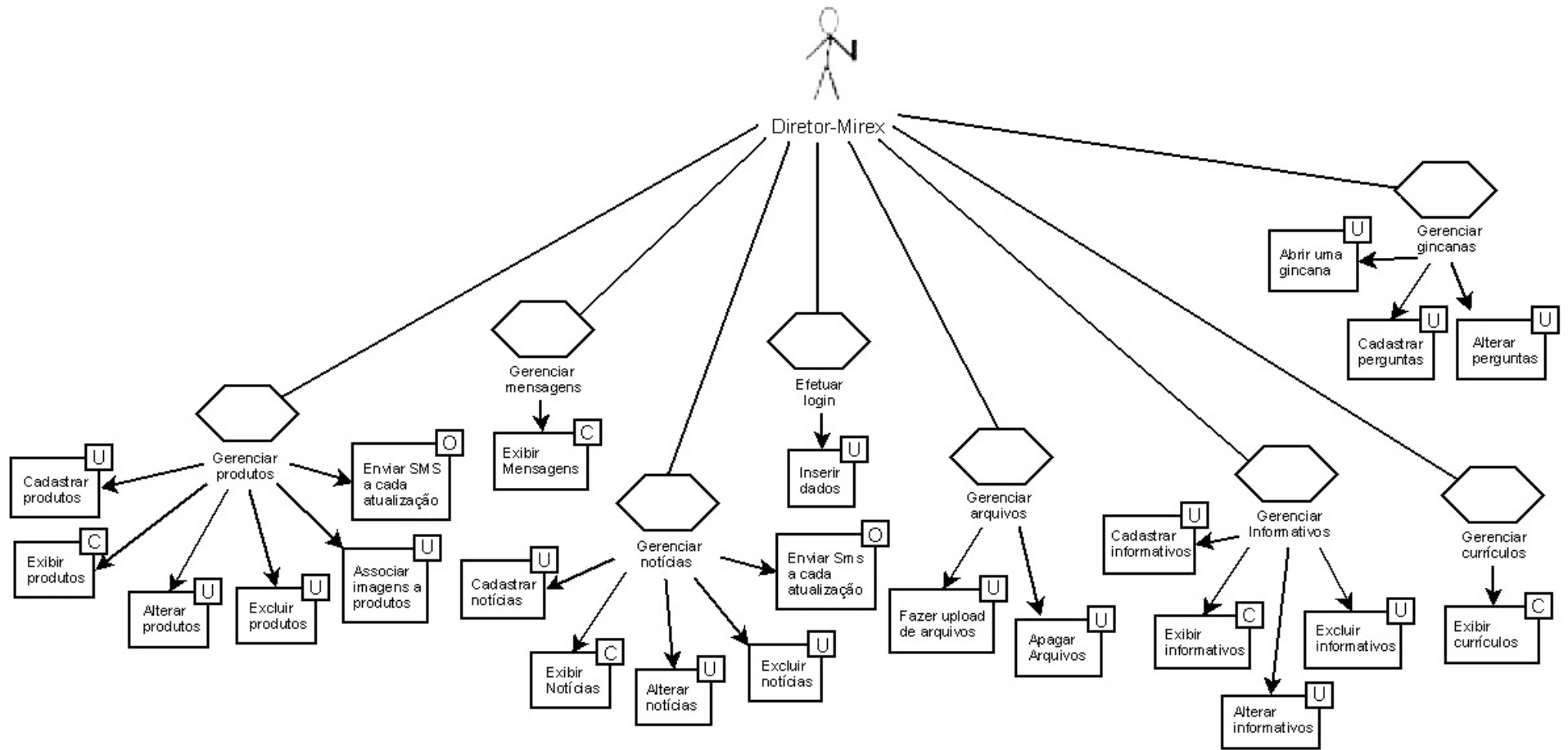


Figura 34: Diagrama GTR para o Diretor da Mirex (Visão de Gerenciamento).

### 5.3.2. Etapa 2: Planejamento Inicial

De acordo com as reuniões e com os diagramas GTR iniciais prontos, nessa etapa foi possível prever duas atividades básicas que deveriam ser conduzidas em paralelo às atividades técnicas de modelagem. São elas: análise de material enviado pela empresa e busca e análise de sites de empresas concorrentes.

O objetivo da primeira atividade foi analisar todo o conteúdo – textual ou não-textual – enviado pela Mirex, conforme foi estipulado nas primeiras reuniões. A partir dessa atividade, foi possível prever também a quantidade de *links* para *download* de arquivos que deveriam ser disponibilizados na aplicação. Já a segunda atividade foi conduzida para avaliar *sites* Web de empresas concorrentes e, assim, identificar fatores e características de apresentação que poderiam ser levadas em consideração durante o desenvolvimento da aplicação Web. No momento em que essas duas atividades foram definidas, o tempo para cada uma delas não foi previsto.

## 5.4. Aplicação da Fase II

Na Fase II, de acordo com o método Web-SEMP, conduziram-se, passo a passo, as três etapas definidas pelo método. Assim, a visão pública foi submetida às Etapas 1, 2 e 3 uma vez que é a visão da aplicação cujo principal objetivo é fornecer conteúdo e possibilidades de navegação aos usuários finais. Já a visão privada (gerenciamento de conteúdos) é formada essencialmente por aspectos funcionais, portanto, as Etapas 1 e 2, para essa visão, não foram seguidas tão rigorosamente como foram para as outras duas visões, mas apenas a Etapa 3. Como a visão privada voltada para balconistas possuía um número muito baixo de requisitos, esta não foi submetida à Fase II do método Web-SEMP.

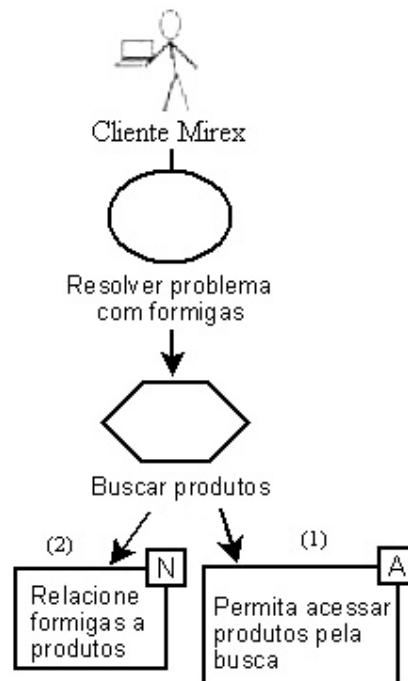
### 5.4.1. Etapa 1: Identificação de Aspectos de Composição

Conforme previsto pelo método Web-SEMP, os passos 1 a 6, os quais compõem a Etapa 1, foram aplicados para a visão pública da aplicação Web.

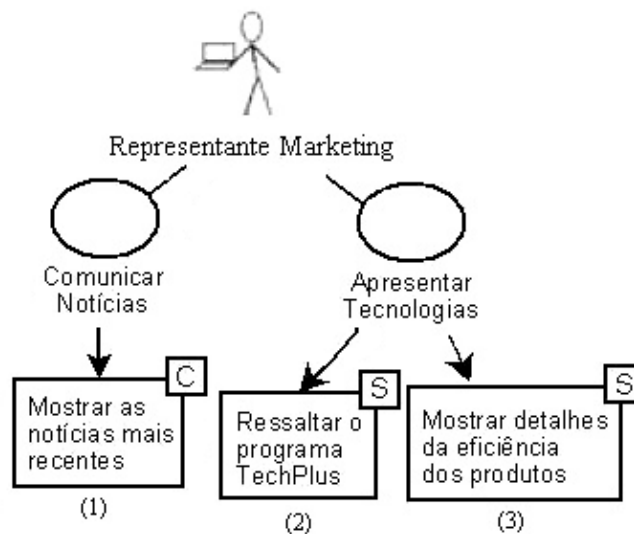
**Visão Pública**

**Passo 1:**

Nesse passo, todos os requisitos de hipermídia foram numerados em cada um dos diagramas GTR relacionados à visão pública: diagramas GTR para o diretor da Mirex, para o representante de *marketing* e para o cliente Mirex. As Figuras 35, 36 e 37 mostram os diagramas numerados.



**Figura 35: Diagrama GTR Numerado para o Cliente Mirex.**



**Figura 36: Diagrama GTR Numerado para o Representante de Marketing.**



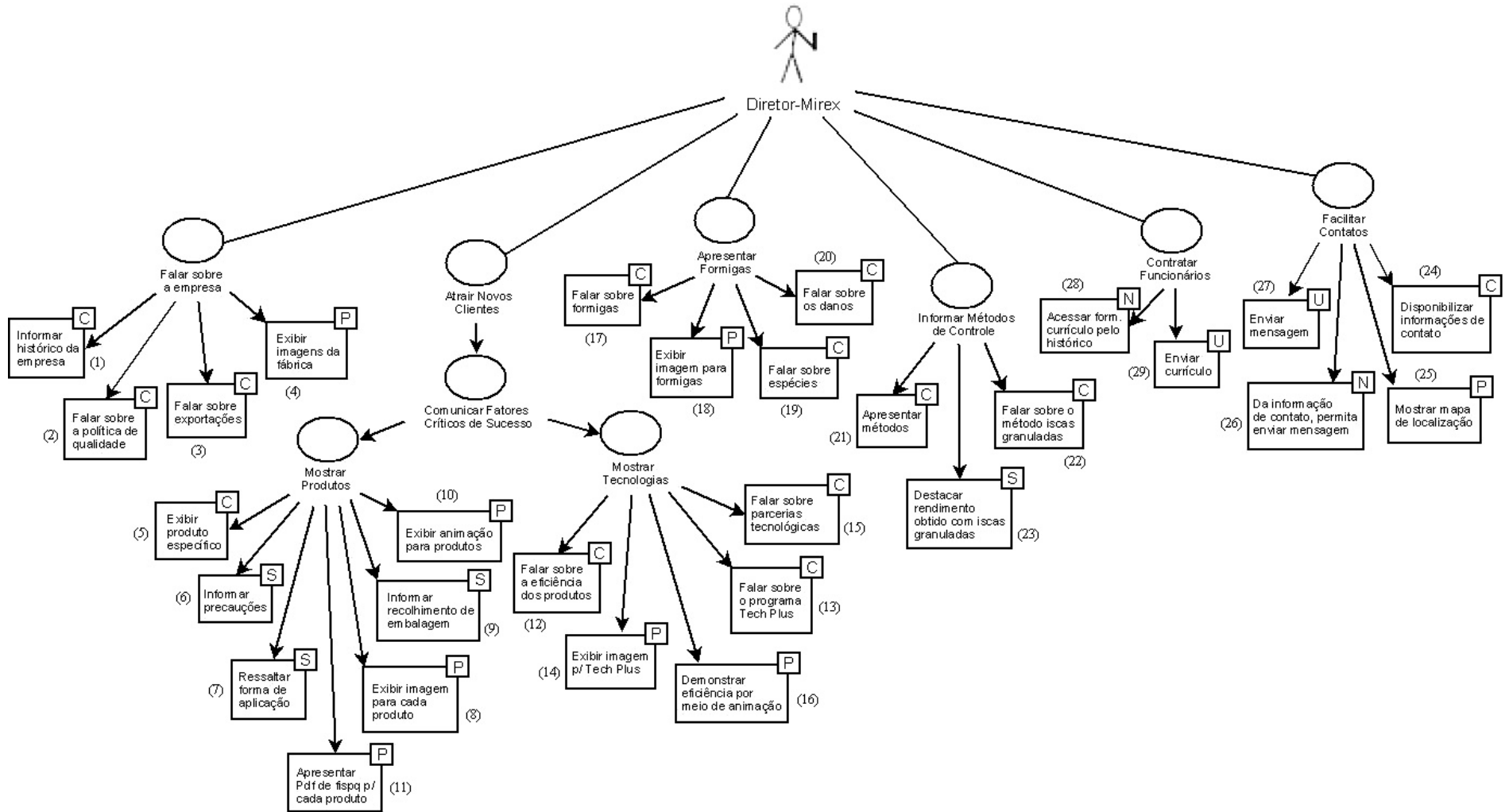


Figura 37: Diagrama GTR Numerado para o Diretor da Mirex.

**Passo 2:**

No Passo 2, identificaram-se todos os candidatos a temas, analisando-se substantivos e cláusulas substantivas que constam em requisitos de conteúdo e de estrutura de conteúdo em todos os diagramas GTR da visão pública. A partir dessa análise, os seguintes temas foram identificados: empresa, produto, tecnologia, formiga, métodos de controle, notícia e contato. A coluna “E/D” foi preenchida com o valor “E” para temas que foram classificados como estáticos e “D” para temas classificados como dinâmicos. A Tabela 17 mostra o formulário TRS preenchido com todas as informações para a Mirex.

**Tabela 17: Formulário TRS Preenchido para a Visão Pública da Mirex.**

<b>Tema</b>	<b>Requisitos Relacionados</b>	<b>Stakeholders</b>	<b>E/D</b>
Empresa	1,2,3,4	Diretor	E
Produto	5,6,7,8,9,10,11,12	Diretor	D
	3	Rep. Marketing	
	1,2	Cliente Mirex	
Tecnologia	12,13,14,15	Diretor	E
	2,3	Rep. Marketing	
Formiga	16,17,18,19	Diretor	E
	2	Cliente Mirex	
Métodos de controle	20,21,22	Diretor	E
Notícia	1	Rep. Marketing	D
Contato	23,24,25,26	Diretor	E

Os passos 3 a 6 foram repetidos para cada tema listado no formulário TRS, conforme é mostrado a seguir.

**Tema: Empresa****Passo 3:**

Cada requisito de conteúdo referente ao tema “Empresa” foi mapeado em uma *text unit*. Uma página principal foi criada para o tema, contendo informações sobre o histórico da empresa. Uma página diferente foi criada para exibir informações sobre a política de qualidade e outra para falar sobre exportações. Ambas as páginas foram

conectadas à página principal do tema, por meio de um *link* não-contextual, conforme mostra a Figura 38.

**Passo 4:**

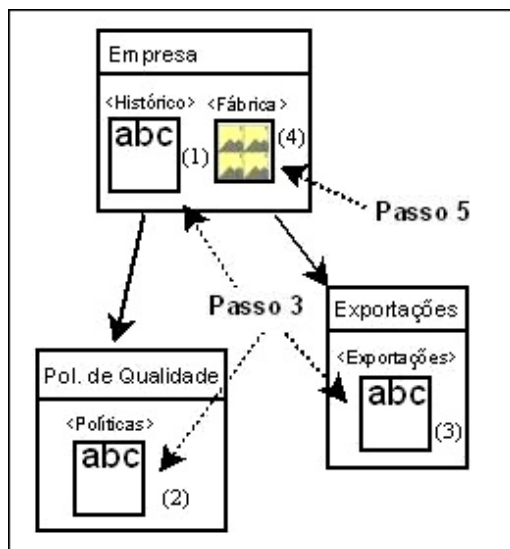
Não há requisitos de estrutura de conteúdo.

**Passo 5:**

Foi verificado um requisito de apresentação expressando a necessidade de exibir imagens da empresa Mirex. Esse requisito foi mapeado em uma *multiimage unit* na página principal do tema, de acordo com a Figura 38.

**Passo 6:**

Não há requisitos de interação.



**Figura 38:** Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Empresa”.

**Tema: Produto****Passo 3:**

Cada requisito de conteúdo – “exibir produto específico” e “falar sobre a eficiência dos produtos” foi mapeado em uma *data unit*, em páginas diferentes. Uma página principal foi criada para o tema, de forma a dar acesso tanto à página contendo

informações de um produto específico quanto à página contendo informações sobre a sua eficiência. A Figura 39 mostra esse mapeamento.

**Passo 4:**

Os três requisitos de estrutura de conteúdo – “informar precauções”, “informar recolhimento de embalagem” e “ressaltar forma de aplicação” foram mapeados, cada um, em uma *data unit* referente a produtos, em páginas diferentes. Como essas informações variam conforme o produto exibido, e precisavam ser colocadas em evidência de alguma forma, as três páginas correspondentes a essas informações foram conectadas à página principal referente ao tema “Produto”, conforme pode ser observado na Figura 39.

**Passo 5:**

O requisito de apresentação “exibir imagens para cada produto” foi mapeado em uma *multiimage unit* conectada à *data unit* responsável por publicar os dados de cada produto. Outro requisito de apresentação – “exibir animação para produtos” foi mapeado em uma *animation unit*, disposta na página principal do tema (contexto estático). Esse requisito, de fato, expressa a necessidade de criar um efeito animado no momento de exibição da lista de produtos na página principal, conforme pode ser observado na Figura 39.

**Passo 6:**

Para o tema “Produto”, não há requisitos de interação.

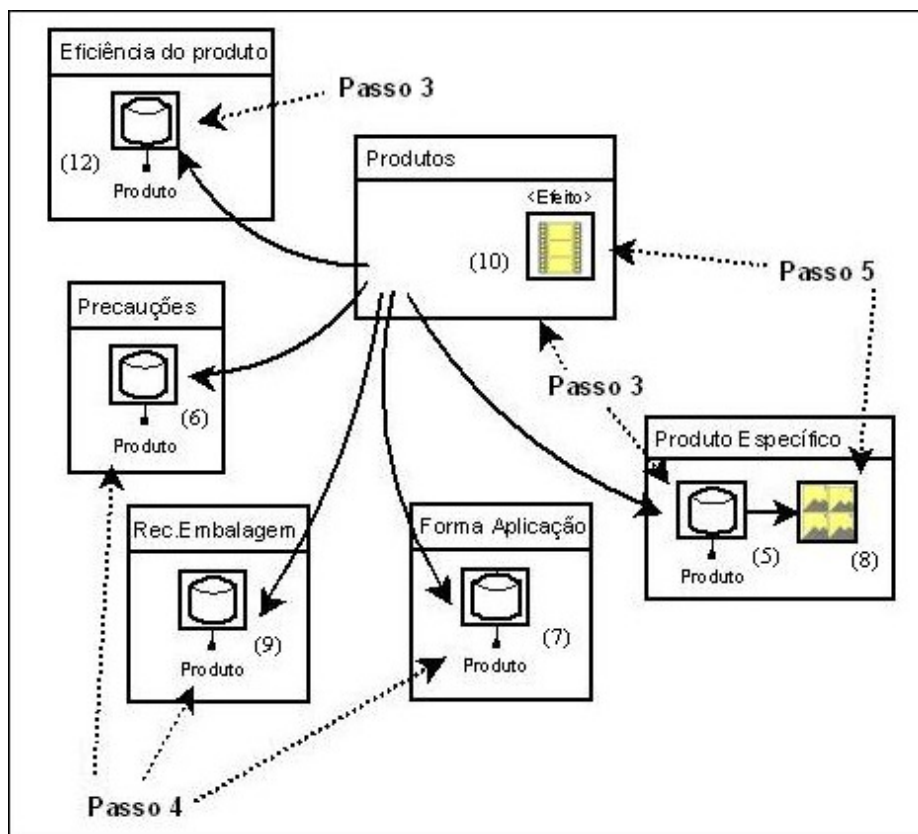


Figura 39: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Produto”.

## Tema: Tecnologia

### Passo 3:

Cada requisito de conteúdo referente ao tema “Tecnologia” foi mapeado em uma *text unit*. Uma página principal foi criada para o tema, contendo informações gerais sobre as tecnologias utilizadas – representadas em uma nova *text unit*. Informações sobre parcerias tecnológicas e sobre programa TechPlus foram dispostas em novas páginas. O requisito de conteúdo “falar sobre a eficiência dos produtos” já havia sido satisfeito quando se tratou o tema “produto”. A Figura 40 mostra esse mapeamento.

### Passo 4:

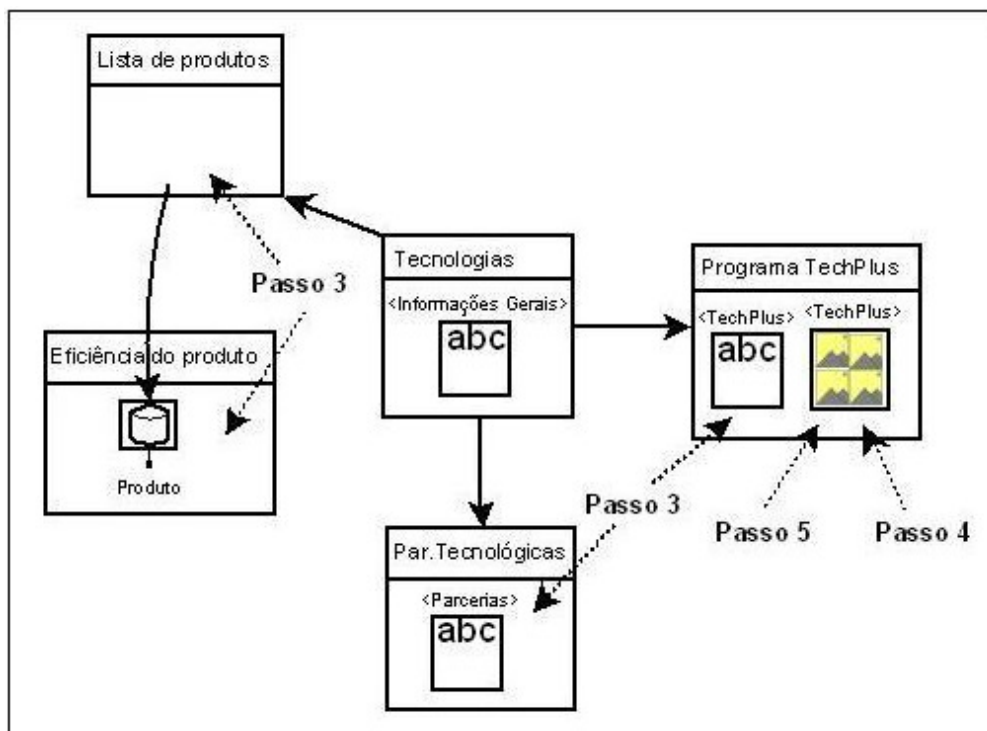
Para o tema “Tecnologia”, há dois requisitos de estrutura de conteúdo: um relacionado ao programa TechPlus e outro relacionado à eficiência dos produtos fornecidos. Para satisfazer o primeiro requisito, foi decidido apresentar imagens para ressaltar o programa, e para satisfazer o segundo, foi decidido apresentar todos os detalhes necessários (em forma de texto) para evidenciar a eficiência das tecnologias usadas nos produtos para combater formigas, conforme pode ser visto na Figura 40.

**Passo 5:**

O requisito de apresentação “exibir imagens para o programa TechPlus” foi mapeado em uma *multiimage unit* disposta na mesma página contendo as informações sobre o programa TechPlus, de acordo com a Figura 40.

**Passo 6:**

Para o tema “Tecnologia”, não há requisitos de interação.



**Figura 40: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Tecnologia”.**

**Tema: Formiga****Passo 3:**

Cada requisito de conteúdo referente ao tema “Formiga” foi mapeado em uma *text unit*. Uma página principal foi criada para o tema, contendo informações sobre formigas. Uma página foi criada para exibir informações sobre danos causados por formigas e outra para falar sobre espécies. Ambas as páginas foram conectadas à página principal do tema, de acordo com a Figura 41.

**Passo 4:**

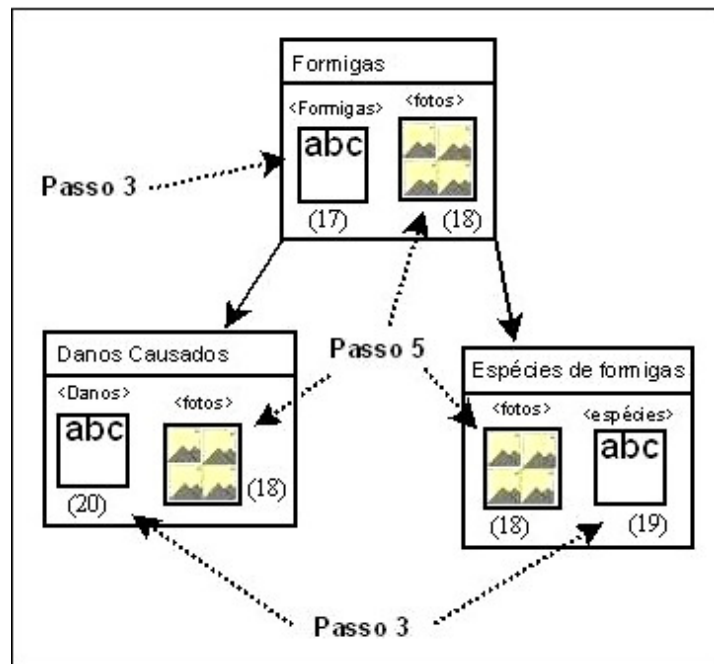
Não há requisitos de estrutura de conteúdo.

**Passo 5:**

O requisito de apresentação “exibir imagens para formigas” foi mapeado em uma *multiimage unit* disposta em cada página correspondente ao tema “Formiga”, conforme a Figura 41.

**Passo 6:**

Não há requisitos de interação.



**Figura 41: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Formiga”.**

**Tema: Métodos de Controle****Passo 3:**

Cada requisito de conteúdo referente ao tema “Métodos de Controle” foi mapeado em uma *text unit*. Uma página principal foi criada para o tema, contendo informações sobre métodos, e outra página, ligada à página principal, foi criada para apresentar informações sobre o método “iscas granuladas”. A Figura 42 mostra esse mapeamento.

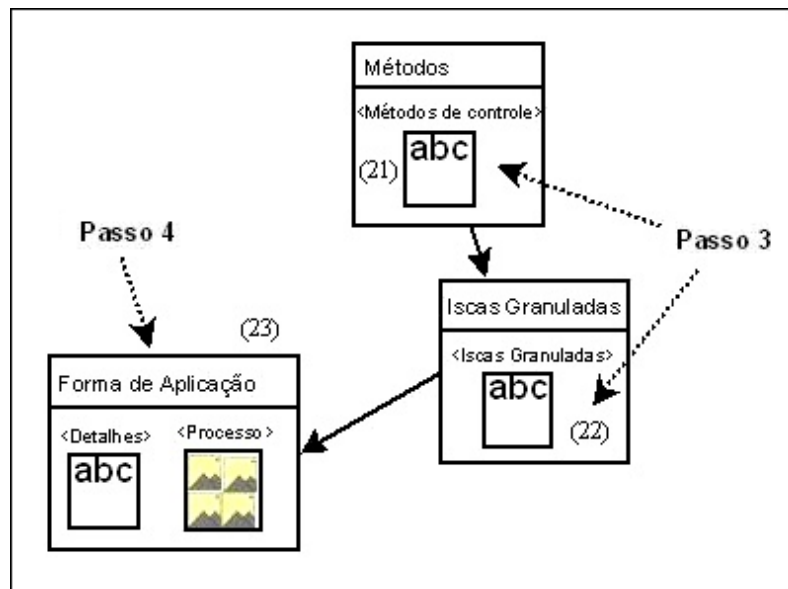
**Passo 4:**

Para satisfazer o requisito de estrutura de conteúdo “destacar o rendimento obtido com iscas granuladas”, decidiu-se detalhar, passo a passo, a forma de aplicação de tal método e apresentar imagens ilustrando esse processo. Portanto, uma *text unit* e

uma *multiimage unit* foram dispostas em uma nova página, ligada à página responsável pela publicação de informações sobre o método “iscas granuladas”, de acordo com a Figura 42.

**Passo 5 e 6:**

Não há requisitos de apresentação/interação.



**Figura 42: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Métodos”.**

**Tema: Notícia**

**Passo 3:**

O requisito de conteúdo “mostrar as notícias mais recentes” foi mapeado em uma *data unit*. Uma página principal foi criada para o tema, de forma a dar acesso à página contendo informações de uma notícia específica. A Figura 43 mostra esse mapeamento.

**Passos 4, 5 e 6:**

Para o tema “Notícia”, não há nenhum outro requisito, além do requisito de conteúdo já identificado.



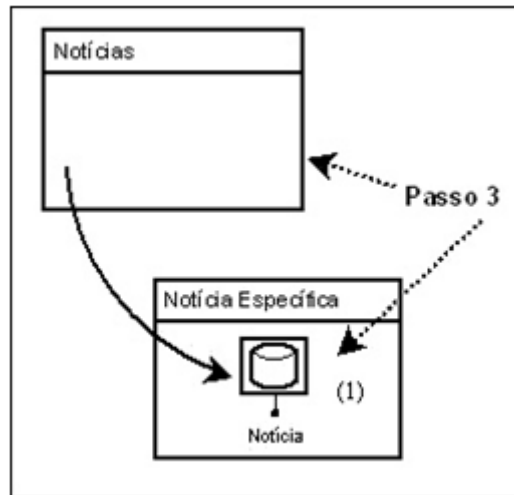


Figura 43: Sub-modelo de Composição após a Aplicação da Etapa 1 para “Notícias”.

### Tema: Contato

#### Passo 3:

O requisito de conteúdo “disponibilizar informações de contato” foi mapeado em uma *text unit*. Uma página principal foi criada para o tema, de forma a dar acesso à página contendo informações de uma notícia específica, conforme é mostrado pela Figura 44.

#### Passos 4:

Não há requisitos de estrutura de conteúdo.

#### Passos 5:

O requisito de apresentação “mostrar mapa de localização” foi mapeado em uma *image unit* para destacar graficamente a localização da empresa, de acordo com a Figura 44.

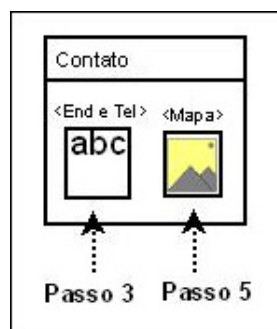


Figura 44: Sub-modelo de Composição Após a Aplicação da Etapa 1 para “Contato”.

## 5.4.2. Etapa 2: Identificação de Aspectos de Navegação

### Visão Pública:

#### Passo 1: Navegação

Foram identificados três requisitos de navegação nos diagramas GTR para a visão pública: “relacione formigas a produtos”, “acessar form. currículo pelo histórico” e “da informação de contato, permita enviar mensagem”. O primeiro requisito indica um relacionamento navegacional entre um tema estático e um tema dinâmico, portanto, foi criado um *link* não-contextual entre as páginas principais dos temas para satisfazer esse requisito. Os outros dois requisitos também devem ser mapeados em *links* não-contextuais permitindo que o usuário envie seu currículo, uma vez tenha acessado a página da empresa, ou envie alguma mensagem, uma vez tenha acessado a página contendo informações sobre contato. A Figura 45 mostra o mapeamento de requisitos de navegação no modelo de hipertexto.

#### Passo2: Caminhos de Acesso ao Conteúdo

Para a visão pública, identificou-se apenas um requisito de caminhos de acesso ao conteúdo: “permita acessar produtos pela busca”. Esse requisito foi resolvido adicionando-se no modelo de hipertexto um formulário de busca – *entry unit* – que deveria ser utilizado em conjunto com um índice – *index unit* – para consultar produtos com base em informações inseridas pelo usuário. Além disso, adicionou-se uma *index unit* na página principal de produtos para dar acesso aos dados de um produto específico. Também, para dar acesso a uma notícia específica, uma *index unit* foi adicionada na página principal do tema “Notícia”. Uma página “Home” foi criada para a visão pública dando acesso a todos os temas anteriormente identificados. A Figura 45 mostra o modelo de hipertexto após a aplicação dos passos 1 e 2.

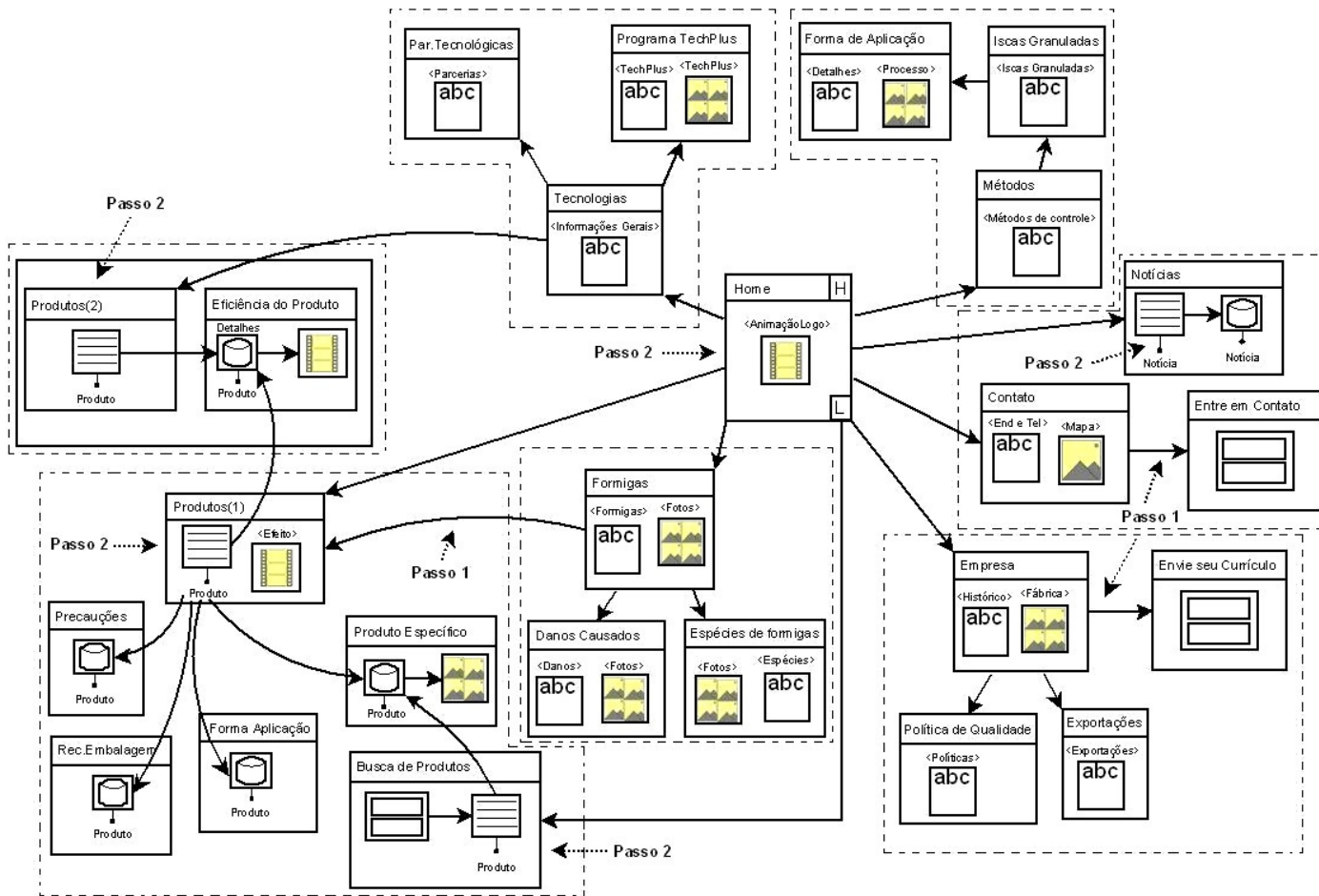


Figura 45: Modelo de Hipertexto após a aplicação dos Passos 1 e 2.

### Passo3: Casos de Uso Navegacionais

A partir da identificação de temas e relacionamentos entre eles, alguns casos de uso navegacionais foram identificados. A Figura 46 mostra o modelo de casos de uso navegacionais. As linhas tracejadas na Figura 45 delimitam cada caso de uso navegacional identificado.

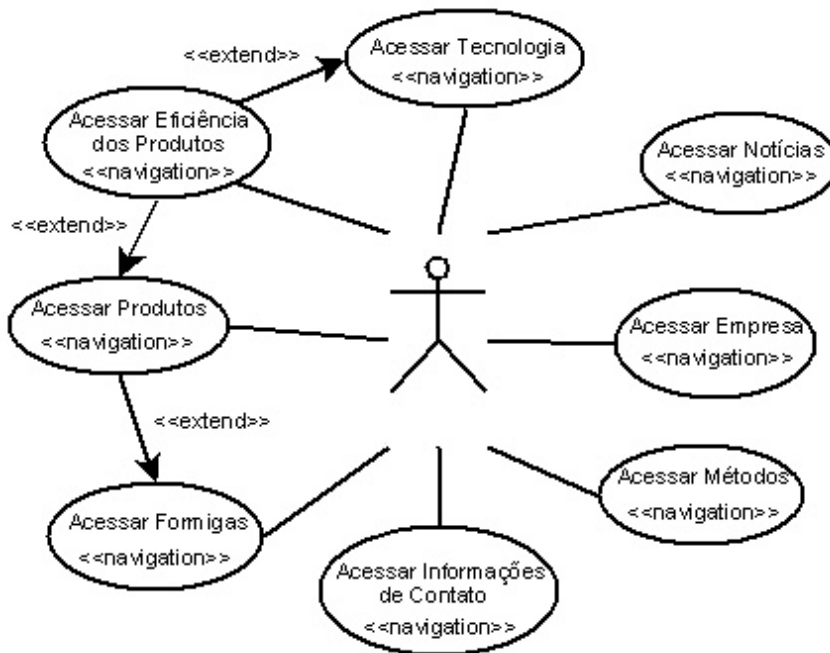


Figura 46: Modelo de Casos de Uso Navegacionais para a Mirex.

### 5.4.3. Etapa 3: Identificação de Aspectos de Funcionalidade

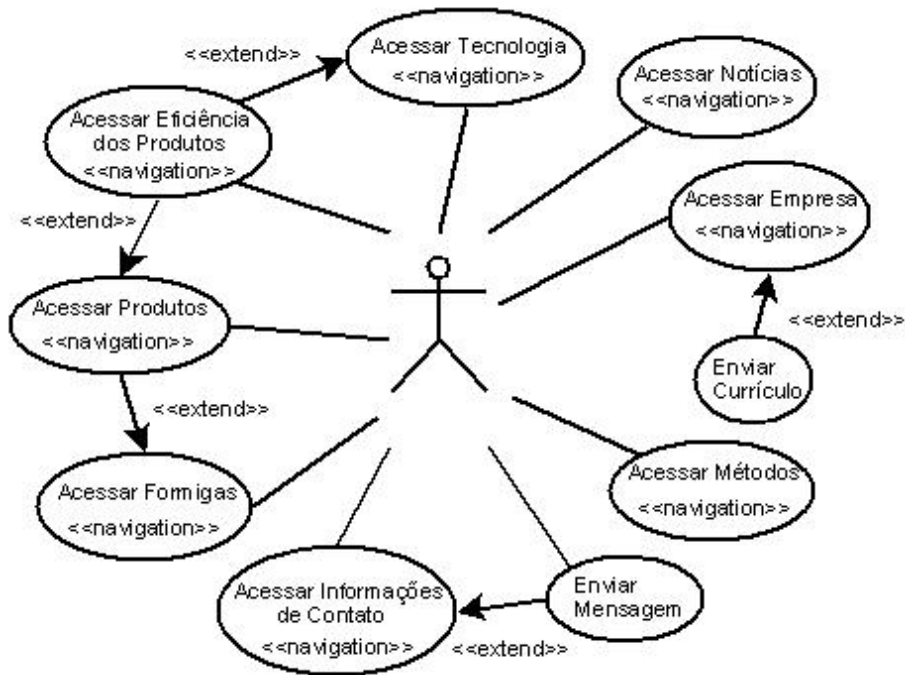
#### Visão Pública:

#### Passo1: Requisitos de operação de usuário

Para a visão pública, dois requisitos de operação de usuário foram identificados: “enviar currículo” e “enviar mensagem”. Ambos os requisitos foram mapeados em casos de uso funcionais. Com base na identificação de temas e relacionamentos entre eles, e na identificação de casos de uso navegacionais na etapa anterior, identificaram-se relacionamentos entre esses dois casos de uso funcionais e os casos de uso navegacionais. A Figura 47 mostra o modelo de casos de uso navegacionais e funcionais.

### Passo2: Requisitos de operação de sistema

Não foi identificado nenhum requisito de operação de sistema.



**Figura 47: Modelo de Casos de Uso Funcionais e Navegacionais para a Visão Pública da Mirex.**

### Visão Privada (Gerenciamento de Conteúdos):

#### Passo1: Requisitos de operação de usuário

Para a visão privada, cada tarefa foi mapeada em um caso de uso funcional. Com exceção das tarefas “gerenciar mensagens”, “efetuar login” e “gerenciar currículos”, em todos os outros casos, cada requisito de operação de usuário no diagrama GTR da Figura 34 foi mapeado em um cenário do caso de uso funcional.

#### Passo2: Requisitos de operação de sistema

O requisito de operação de sistema “enviar SMS a cada atualização” foi mapeado em um caso de uso funcional, associado com os outros casos de uso derivados das tarefas que possuem esse requisito, por meio do estereótipo *<<include>>*.

## 5.5. Aplicação da Fase III

Após a aplicação da Fase II do método Web-SEMP, identificaram-se todos os casos de uso navegacionais e funcionais que compõem a aplicação. Nessa fase, cada caso de uso foi analisado para a avaliação de complexidade, envolvendo tanto características funcionais quanto de hipermídia.

A Fase III do método Web-SEMP ainda está sendo conduzida pela equipe de Engenharia da Web da Linkway e, portanto, nesta seção são apresentados apenas exemplos de como a contagem deveria ser feita para cada visão da Mirex.

### 5.5.1. Etapa 1: Contagem de Caso de Uso Navegacional Estático

Cada caso de uso navegacional estático deveria ser avaliado de acordo com as características de todos os elementos de hipermídia constituindo cada uma das páginas compreendidas pelo caso de uso. Tomando como exemplo o caso de uso “acessar empresa”, existem quatro páginas, conectadas por meio de três *links* não-contextuais, compondo o caso de uso. De acordo com a estratégia de aplicação, cada uma dessas páginas, a partir da página “Empresa”, deveria ser avaliada analisando-se a complexidade de *text units* nas páginas “Empresa”, “Política de Qualidade” e “Exportações” e uma *multiimage unit* na página “Empresa”. O formulário – representado por uma *entry unit* – também deveria ser avaliado.

### 5.5.2. Etapa 2: Contagem de Caso de Uso Navegacional Dinâmico

Cada caso de uso navegacional dinâmico deveria ser avaliado de acordo com a sua complexidade de hipermídia seguindo-se as mesmas regras da Etapa 1 e também de acordo com a sua complexidade funcional, usando-se o método PCU. Tomando como exemplo o caso de uso “Acessar Produtos”, a sua página de entrada – “Produtos(1)” deveria ser avaliada de acordo com a complexidade da animação – *animation unit* – apresentada nesta página. A quantidade de *links* contextuais saindo dessa página deveria ser registrada (4 *links* contextuais), e, então, cada um desses *links* deveria ser seguido para avaliar a complexidade dos elementos de hipermídia que constituem cada uma das páginas de destino. Após essa avaliação, o caso de uso deveria ser avaliado conforme a quantidade de cenários ou objetos de análise, conforme proposto por Karner (1993).

### 5.5.3. Etapa 3: Contagem de Caso de Uso Funcional

Para a visão pública, os dois casos de uso funcionais – “Enviar Currículo” e “Enviar Mensagem” deveriam ser avaliados usando-se apenas o método PCU, comentado no Capítulo 2. A mesma regra se aplica a todos os casos de uso funcionais identificados para a visão privada (gerenciamento de conteúdos) da Mirex.

### 5.5.4. Etapa 4: Avaliação de Complexidade Total da Aplicação Web

Nessa etapa, os valores de pontos atribuídos a cada caso de uso navegacional – seja estático ou dinâmico – deveriam ser somados, assim como a quantidade de *links* e páginas envolvidas por um caso de uso. O número de *links* e páginas servirão de entrada para avaliar a complexidade estrutural de uma aplicação Web, conforme proposto pela métrica WHP, apresentada no Capítulo 3. Assim, nessa etapa, obtém-se um valor de pontos referente à complexidade de hipermídia da aplicação e um valor de pontos referente a sua complexidade funcional.

## 5.6. Considerações Finais

Neste capítulo, apresentou-se um estudo de caso relatando a aplicação do método Web-SEMP em uma aplicação real desenvolvida pela empresa mencionada neste trabalho. Os resultados obtidos mostraram que o método deu apoio efetivo às atividades de elicitação, modelagem e planejamento para aplicações Web e que os modelos construídos facilitaram a comunicação entre todos os envolvidos no projeto, independentemente do seu conhecimento técnico. Embora o método Web-SEMP tenha exigido um treinamento inicial para ser adotado, não se encontraram dificuldades durante a sua aplicação e as suas representações permitiram que todos os *stakeholders* discutissem em torno dos modelos, proporcionando uma definição de requisitos mais completa e mais condizente com as necessidades deles.

A partir dos modelos de análise construídos, foi possível determinar as correspondências entre os elementos contidos nos modelos de hipertexto e nos modelos de objetivos e requisitos. Assim, a equipe de desenvolvimento pôde controlar e rastrear requisitos e modificações de requisitos em qualquer momento, durante a realização do projeto. Por meio dessas correspondências, foi possível entender rapidamente como a modificação em um objetivo ou requisito poderia afetar diferentes aspectos do projeto em desenvolvimento. Conforme citado anteriormente, os resultados da aplicação da

Fase III não foram apresentados, uma vez que a atividade de planejamento ainda está sendo conduzida pela equipe de desenvolvimento e não se pode concluir que as estimativas feitas com base nos modelos estejam condizentes com o tempo real consumido durante o desenvolvimento.



# CAPÍTULO 6

## CONCLUSÕES

---

Este trabalho apresentou o método Web-SEMP, cujo objetivo é dar apoio efetivo às atividades de elicitação, modelagem e planejamento de aplicações Web. Além disso, este trabalho apresentou a métrica WHP, especificamente projetada para avaliar a complexidade de elementos de hipermídia que constituem uma aplicação Web.

O método Web-SEMP foi definido por observarem-se, na literatura, algumas lacunas tais como: i) dentre os diversos métodos analisados para dar suporte à fase de Engenharia de Requisitos para Web, não se identificou nenhum que tratasse essa fase como um todo, ou seja, que apresentasse uma proposta que desse suporte à elicitação de requisitos e que propusesse uma migração bem definida para uma fase de modelagem desses requisitos, tanto no que diz respeito aos aspectos navegacionais como aos aspectos funcionais; ii) dentre as diversas técnicas para determinação de esforço de desenvolvimento de aplicações Web, não se identificou nenhuma que tratasse de propriedades específicas de alguns elementos de hipermídia que possuem um impacto significativo na estimativa de tempo de desenvolvimento de tais aplicações.

Assim, para suprir essas lacunas, o método Web-SEMP, composto por três fases, agrega métodos propostos na literatura, os quais possuem, individualmente, características apropriadas para tratar certos aspectos de aplicações Web.

A Fase I corresponde às atividades de concepção, levantamento e especificação de requisitos. Para essas atividades, o método proposto adota o modelo AWARE, uma abordagem orientada a objetivos que é especialmente adequada para o domínio Web, pois permite tratar, de maneira abrangente, necessidades vagas e objetivos imprecisos, a partir dos quais o desenvolvimento de aplicações Web normalmente é iniciado. Na Fase I, ainda é possível que, através da atividade de Planejamento Inicial, analistas e engenheiros da Web façam uma previsão de certos tipos de tarefas não-técnicas que deverão ser conduzidas posteriormente no processo de desenvolvimento.

A Fase II do método Web-SEMP propõe um conjunto de diretrizes para auxiliar na construção mais sistemática e padronizada de modelos conceituais de projeto, focando, particularmente, no modelo de hipertexto. A partir da Fase II, também é possível identificar as entidades e relacionamentos mais relevantes do modelo de dados, além de casos de uso funcionais e navegacionais. Para essa fase, foram adotados dois métodos: 1) WebML, que dá apoio à elaboração dos modelos de dados e de hipertexto, mas que foi estendido para permitir especificar elementos de hipermídia ainda não considerados, os quais são importantes tanto para a fase de elicitação quanto para a fase de planejamento; 2) Modelo de Casos de Uso Funcionais e Navegacionais, pois este permite representar os dois enfoques – hipermídia e funcionalidade – que podem ser encontrados em uma aplicação Web.

A Fase III corresponde à aplicação do modelo de métrica a todos os modelos conceituais construídos na fase anterior. Assim, devem-se contabilizar os aspectos mais relevantes que compõem uma aplicação Web, considerando-se tanto elementos funcionais quanto de hipermídia. Para essa fase, o método PCU é adotado para avaliar a complexidade funcional, e a métrica WHP, definida neste trabalho, é adotada para avaliar a complexidade de hipermídia de aplicações Web.

A métrica proposta foi definida a partir de um estudo realizado em uma pequena empresa de desenvolvimento de software da cidade de São Carlos. A métrica, estabelecida a partir de dados históricos de 15 aplicações Web, considera propriedades específicas de elementos de hipermídia que compõem uma aplicação Web, aprimorando assim técnicas para o cálculo de esforço propostas na literatura.

Este trabalho também apresentou um estudo de caso para colocar em prática os principais conceitos estabelecidos pelo método Web-SEMP, de forma a possibilitar uma primeira avaliação de sua efetividade, e os resultados fornecem indícios de que o objetivo de manter sob controle, em um único processo, as atividades de análise e projeto de requisitos é alcançado, contribuindo efetivamente para a melhoria da qualidade do processo de desenvolvimento de aplicações Web.

## 6.1. Contribuições do Trabalho

A seguir relacionam-se, resumidamente, algumas contribuições deste trabalho:

- Definição do método Web-SEMP, que tem como objetivo principal fornecer um processo guiado para auxiliar nas atividades de elicitação, modelagem e planejamento para Web, estabelecendo passos para a condução de cada uma dessas atividades.
- Estabelecimento de diretrizes para a construção efetiva de modelos conceituais de projeto, com ênfase no modelo de hipertexto de uma aplicação Web.
- Auxílio na atividade de validação de requisitos, uma vez que, baseando-se em objetivos mais genéricos, o método permite derivar modelos, logo no início da Fase de ER, que tornem mais efetiva a comunicação entre todos os *stakeholders* envolvidos no projeto, possibilitando um feedback mais concreto.
- Auxílio na atividade de gerenciamento de requisitos, na medida que, a partir da aplicação das diretrizes propostas pelo método, é possível manter objetivos e requisitos amarrados a aspectos cruciais de projeto facilitando o entendimento sobre como a modificação em um requisito afetará diferentes aspectos da aplicação Web a ser construída.
- Definição da métrica WHP para avaliar a complexidade de elementos de hipermídia considerando-se propriedades específicas de tais elementos de forma a possibilitar a geração de modelos de estimativa de esforço mais precisos.
- Definição de um modelo de métrica capaz de avaliar, por meio da união de duas métricas – WHP e PCU –, todos os aspectos relevantes relacionados a características funcionais e de hipermídia de uma aplicação Web.
- Extensão do conjunto de unidades de conteúdo definidas pela WebML fornecendo condições de representar, em nível conceitual, aspectos de conteúdo fundamentais no domínio Web, tanto no que diz respeito à atividade de elicitação de requisitos quanto à atividade de planejamento.

## **6.2. Trabalhos Futuros**

A seguir relacionam-se, resumidamente, algumas propostas de trabalhos futuros:

- Automatizar cada uma das fases que compõem o método Web-SEMP, dando apoio ferramental necessário para facilitar a sua aplicação.
- Utilizá-lo em mais aplicações Web a fim de avaliar mais precisamente sua efetividade.
- Aprimorar a métrica proposta com base na coleta de uma quantidade maior de dados históricos.
- Efetuar comparações entre a métrica WHP e outras métricas propostas na literatura.
- Planejar e realizar experimentos inicialmente no meio acadêmico para avaliar e melhor caracterizar o método aqui proposto.

# REFERÊNCIAS BIBLIOGRÁFICAS

---

- [Albrecht, 1979] Albrecht A.J., Measuring Application Development Productivity, in *proceeding IBM Applications Development Symposyum*, IBM Corporation, Monterey, Canada, 1979, pp.14-17.
- [Albrecht & Gaffney, 1983] Albrecht, A. J.; Gaffney, J. R. Software function, source lines of code, and development effort prediction: a software science validation. *IEEE Trans. Software Eng*, 1983, pp.639-648.
- [Al-Salem & Samaha, 2007] Al-Salem, L. S.; Samaha, A.B. Eliciting Web application requirements - an industrial case study. *J. Syst. Softw.*2007, pp.294-313.
- [Al-Subaie & Maibaum, 2006] Al-Subaie, H.; Maibaum, T., Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Methodology, *IEEE International Workshop on Comparative Evaluation in Requirements Engineering (CERE)*, Los Alamitos, California: IEEE Computer Society Press, 2006, pp.8-19.
- [Anton & Potts, 1998] Antón, A. I.; Potts, C. The use of goals to surface requirements for evolving systems. In *Proceedings of the 20<sup>th</sup> international Conference on Software Engineering*. International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 1998, pp.157-166.
- [Baresi et al., 2001] Baresi, L., Garzotto, F., Paolini, P., Extending UML for Modeling Web Applications, *Proc. Of the 34<sup>th</sup> Hawaii International Conference on Systems Sciences (HICSS 2001)*, Maui, HI, Jan 2001, pp.615-626.
- [Baresi et al., 2006] Baresi, L.; Colazzo, S.; Mainetti, L.; Morasca, S. W2000: A Modeling Notation for Complex Web Applications. In *E. Mendes and N. Mosley (eds.) Web Engineering: Theory and Practice of Metrics and Measurement for Web Development*. Springer, 2006.
- [Bleisten et al., 2004] Bleistein, S., et al., Modeling Business Strategy in E-Business Systems Requirements Engineering, *Lecture Notes in Computer Science*, v. 3289, Berlin: Springer, 2004, pp.617-628.
- [Bolchini & Mylopoulos, 2003] Bolchini, D.; Mylopoulos, J. From Task-Oriented to Goal-Oriented Web Requirements Analysis. In: *Proceedings of the Fourth international Conference on Web information Systems Engineering. WISE*. IEEE Computer Society, Washington, DC, 2003, pp.10-19.
- [Bolchini et al., 2003] Bolchini, D.; Paolini, P.; Randazzo, G. Adding Hypermedia Requirements to Goal-Driven Analysis. In: *Proceedings of the 11th IEEE international Conference on Requirements Engineering*. IEEE Computer Society, Washington, DC, 2003, pp.127-137.

- [Bolchini & Paolini, 2004] Bolchini, D.; Paolini, P. Goal-driven requirements analysis for hypermedia-intensive Web applications. *Requir. Eng.*, 2004, pp.85-103.
- [Carvalho et al., 2006] Carvalho, A.; Chiossi, T.; Drach, M. Aplicabilidade de Métricas por ponto de função a sistemas baseados em web. In: *Anais do WER06 - Workshop em Engenharia de Requisitos*, Rio de Janeiro, Brasil. 2006, pp.109-115.
- [Castro et al., 2002] Castro, J.; Kolp, M.; Mylopoulos, J. *Towards requirements driven information systems engineering: the TROPOS project*. *Inform Syst.*, 2002, pp.365-389.
- [Ceri et al., 2000] Ceri, S.; Fraternali, P.; Bongio, A. Web Modeling Language (WebML): a modeling language for designing Web sites. In: *Proceedings of the 9th international World Wide Web Conference on Computer Networks: the international Journal of Computer and Telecommunications Netowrking. North-Holland Publishing Co.*, Amsterdam, The Netherlands, 2000, pp.137-157.
- [Ceri et al., 2002] Ceri, S.; Fraternali, P.; Matera, M. *Conceptual Modeling of Data-Intensive Web Applications*. *Internet Computing*, 2002, pp.20-30.
- [Ceri et al., 2003] Ceri, S; Daniel, F.; Matera, M. Extending WebML for modeling multi-channel context-aware web applications. In: *Proceedings of Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03)*, Rome, Italy, IEEE Press, 2003, pp.225-233.
- [Cleary, 2000] Cleary, D. Web-Based Development and Functional Size Measurement. In: *Proceedings of IFPUG Annual Conference*, San diego, USA. 2000.
- [Conallen, 2003] Conallen, J. *Building Web Applications With UML*, 2nd ed. Reading, MA: Addison-Wesley, 2003, 496p.
- [Conte et al., 1986] Conte, S.; Dunsmore, H.; Shen, V. *Software Engineering Metrics and Models*. Benjamin/Cummings, Menlo Park, California, 1986, 396p.
- [Dardenne, et al., 1993] Dardenne, A.; van Lamsweerde, A.; Fickas, S. Goal-directed requirements acquisition. In: *Selected Papers of the Sixth international Workshop on Software Specification and Design* M. Sintzoff, C. Ghezzi, and G. Roman, Eds. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1993, pp.3-50.
- [Deshpande & Hansen, 2001] Deshpande, Y; Hansen, S. Web Engineering: Creating a Discipline among Disciplines. *IEEE MultiMedia*, 2001, pp.82-87.
- [Deshpande et al., 2002] Deshpande, Y.; Murugesan, S.; Ginige, A.; Hansen, S.; Schwabe, D.; Gaedke, M.; White, B. Web Engineering. *Journal of Web Engineering*, 2002, pp.3-17.

- [Escalona & Koch, 2006] Escalona, M. J.; Koch, N. Metamodeling the Requirements of Web Systems. In: *WEBIST 2006- The Second International Conference for Web Information Systems and Technologies*, Setubal, Portugal, 2006, pp.310-317.
- [Garzotto & Paolini, 1993] Garzotto, F.; Paolini, P. A model-based approach to hypertext application design. *ACM Trans. Inform. Syst.*, 1993, pp.1-26.
- [Gordijn & Akkermans, 2003] Gordijn, J.; Akkermans, J. M. Value-based requirements engineering: exploring innovative e-commerce ideas. *Requir. Eng.*, 2003, pp.114-134.
- [IFPUG, 1998] IFPUG, Function Point Counting Practices Manual, Release 4.1, *International Function Point Users Group*, Westerville, Ohio, USA, 1998.
- [Jacobson et al., 1992] Jacobson, I.; Christerson, M.; Jonsson, P.; Overgaard G. *Object-Oriented Software Engineering – A Use Case Driven Approach*. Addison-Wesley Publish Company, 1992, 528p.
- [Kappel et al., 2006] Kappel, G., Proll, B, Reich, S., and Retschitzegger, W. *Web Engineering: The Discipline of Systematic Development of Web Applications*. John Wiley & Sons, 2006, 366p.
- [Karner, 1993] Karner, G. *Resource Estimation for Objectory Projects*, Objective Systems SF AB, Set.,1993.
- [Koch & Kraus, 2002] Koch, N.; Kraus, A. The Expressive Power of UML-Based Web Engineering. In: *Proc. of 2<sup>nd</sup> International Workshop on Web-Oriented Software Technology*, Malaga, Spain, 2002.
- [Leffingwell & Widrig, 2003] Leffingwell, D; Widrig, D. *Managing Software Requirements: a Use Case approach*. 2. ed. Addison-Wesley, 2003, 550p.
- [Martino et al., 2007] S. D. Martino, F. Ferrucci, L. Paolino, M. Sebillio, G. Vitiello, and G. Avagliano. A WebML-based Visual Language for the Development of Web GIS Applications. In: *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing. VLHCC*. IEEE Computer Society, Washington, DC, Set. 2007, pp.209-214.
- [McDonald & Welland, 2001] Welland, R.; McDonald, A. Web Engineering in Practice. In: *Proceedings of the 4<sup>th</sup> WWW10 Workshop on Web Engineering*. 2001, pp.21-30.
- [Meldrum & Rose, 2004] Meldrum, M.; Rose, J. Activity based generation of requirements for Web-based information systems: the SSM/ICDT approach. In: *The 12th European Conference on Information Systems (ECIS 2004)*, Turku Finland, 2004.

- [Mendes et al., 2001] Mendes, E.; Watson, I.; Triggs, C.; Mosley, N.; Counsell, S. A comparison of length, complexity and functionality as size measures for predicting web design and authoring effort. In *Proceedings of the IEEE Metrics Symposium*. IEEE Computer Society Press, Los Alamitos, CA. 2001.
- [Mendes et al., 2002] E. Mendes, N. Mosley, and S. Counsell, Comparison of Web size measures for predicting Web design and authoring effort, *IEE Proceedings - Software*, v. 149, n. 3, 2002, pp. 86-92.
- [Mendes et al., 2003] Mendes, E., Mosley, N., and Counsell, S. Early Web Size Measures and Effort Prediction for Web Costimation. In: *Proceedings of the 9th international Symposium on Software Metrics*. METRICS. IEEE Computer Society, Washington, DC, Set. 2003.
- [Mendes et al., 2005a] Mendes, E.; Mosley, N.; Counsell, S. Investigating web size metrics for early web cost estimation. *Journal of Systems and Software*, v. 77, n.2: 2005a, pp. 157-172.
- [Mendes et al., 2005b] Mendes, E.; Counsell, S.; Mosley, N. Towards a Taxonomy of Hypermedia and Web Application Size Metrics. In: *Proceedings of 5<sup>th</sup> International Conference on Web Engineering (ICWE 2005)*, Sydney, Australia. Springer. 2005b, pp. 110-123.
- [Perrone & Bolchini, 2004] Perrone, V.; Bolchini, D. Designing Communication Intensive Web Applications: Experience and Lessons from a Real Case. In *proc. of WER 2004 - Tandil, Argentina*. Dez. 2004, pp. 9-10.
- [Perrone et al., 2005a] Perrone, V.; Bolchini, D.; Rastellini, A.; Dragone, L. Shaping Requirements for Institutional Web Applications: Experience from an Industrial Project. In: *Proceedings of the 13th IEEE international Conference on Requirements Engineering (Re'05)*, Washington, DC. IEEE Computer Society. 2005, pp. 221-230.
- [Perrone et al., 2005b] Perrone, V.; Bolchini, D.; Paolini, P. A stakeholders centered approach for conceptual modeling of communication-intensive applications. In *Proceedings of the 23rd Annual international Conference on Design of Communication: Documenting & Designing For Pervasive information*. SIGDOC '05. ACM, New York, NY, 2005, pp. 25-33.
- [Pfleeger, 2004] Pfleeger, S. L. *Engenharia de Software – Teoria e Prática*. 2. ed. Prentice-Hall, 2004. 535 p.
- [Pressman, 2006] Pressman, R. S. *Engenharia de Software*. 6. ed., McGraw-Hill, 2006. 720p.
- [Redouane, 2002] Redouane, A. Experience Using Formal Methods for Capturing Requirements of Web-Based Applications. In: *Proceedings of the 1<sup>st</sup> IEEE international Conference on Cognitive informatics (ICCI)*, Washington, DC. IEEE Computer Society. 2002, pp. 213-221.



- [Redouane, 2004] Redouane, A. Towards a New Method for the Development of Web-Based Applications. In: *Proceedings of the 3<sup>rd</sup> IEEE international Conference on Cognitive informatics (Icci'04)*, Washignton, DC. IEEE Computer Society. 2004, pp. 116-122.
- [Reifer, 2000] Reifer, D. J. Web Development: Estimating Quick-to-Market Software. *IEEE Software*, 2000, pp. 57-64.
- [Reifer, 2002] Reifer, D. Ten deadly risks in internet and intranet software development. *IEEE Software*. v.18, 2002, pp. 12-14.
- [Rollo, 2000] Rollo, T. Sizing e-commerce. In: *Proceedings of the ASCOM 2000*, Sydney, Australia.
- [Rossi et al., 2001] Rossi, G; Schwabe, D; Guimaraes, R. Designing personalized web applications, In: *Proc. WWW10*, Hong Kong, 2001.
- [Sanches et al., 2007] Sanches, A.; Montebelo, R. P.; Fabbri, S. C. P. F. PCUIPSP: Uma Estratégia para ajustar Pontos por Casos de Uso por meio do PSP em Empresas de Pequeno Porte. In: *SBQS 07 Simpósio Brasileiro de Qualidade de Software 2007*, 2007, Porto de Galinhas. Anais do VI Simpósio Brasileiro de Qualidade de Software, 2007. v. 1. pp. 187-202.
- [Yu, 1993] Yu, E. Modeling organizations for information systems requirements engineering. In: *Proc 1st international symposium on requirements engineering, RE'93*, San Jose, USA, 1993.
- [Zhang et al., 2003] Zhang, J.; Chang, C. K.; Chung, J.-Y. Mockup-driven Fast-prototyping Methodology for Web Requirements Engineering. In: *Proceedings of the 27<sup>th</sup> Annual international Conference on Computer Software and Applications (COMPSAC)*, Washington, DC. IEEE Computer Society, 2003.
- [Zhang & Chung, 2003] Zhang, J.; Chung, J.-Y. Mockup-driven Fast-prototyping Methodology for Web Applications. In: *Proceedings of the 2003 Symposium on Applications and the internet (SAINT)*, Washington, DC. IEEE Computer Society. 2003, pp. 410-413.
- [Zhang & Buy, 2003] Zhang, J.; Buy, U. A Framework for the Efficient Production of Web Applications. In: *Proceedings of the 8<sup>th</sup> IEEE international Symposium on Computers and Communications*, Washington, DC. IEEE Computer Society. 2003, pp. 419-424.

# APÊNDICE A

Neste apêndice apresentam-se as tabelas com os valores de pontos atribuídos aos elementos de hipermídia de cada site Web usado para derivar a métrica WHP.

Site: Reiseg (Sistemas contra incêndio)

Endereço: [www.reiseg.com.br](http://www.reiseg.com.br)

Elementos de Hipermídia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	0	Baixa	x 30	0	0
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	1	Baixa	x 30	30	30
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Text Unit</i>	3	Baixa	x 1	3	3
	0	Média	x 2	0	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	120	Baixa	x 2	240	240
	0	Média	x 6	0	
	0	Alta	x 12	0	
<i>Animation Unit</i>	1	Baixa	x 15	15	15
	0	Média	x 60	0	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural*	300				
<b>Total de pontos: 588</b>					

\* Os valores de pontos possíveis para avaliar a complexidade estrutural de uma aplicação Web são: 100 pontos (baixa), 300 pontos (média) e 550 pontos (alta).

Site: O2Personal (Academia)

Endereço: www.o2personal.com.br

Elementos de Hiperímia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	0	Baixa	x 30	0	0
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	1	Baixa	x 30	30	30
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Text Unit</i>	19	Baixa	x 1	19	19
	0	Média	x 2	0	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	86	Baixa	x 2	172	256
	14	Média	x 6	84	
	0	Alta	x 12	0	
<i>Animation Unit</i>	3	Baixa	x 15	45	45
	0	Média	x 60	0	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	300				
<b>Total de pontos: 650</b>					

Site: Sorts (Agência de turismo)

Endereço: www.sorts.com.br

Elementos de Hiperímídia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	0	Baixa	x 30	0	0
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	1	Baixa	x 30	30	210
	2	Média	x 50	100	
	1	Alta	x 80	80	
<i>Text Unit</i>	1	Baixa	x 1	1	1
	0	Média	x 2	0	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	0	Baixa	x 2	0	132
	0	Média	x 6	0	
	11	Alta	x 12	132	
<i>Animation Unit</i>	0	Baixa	x 15	0	130
	0	Média	x 60	0	
	1	Alta	x 130	130	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	100				
<b>Total de pontos: 573</b>					

Site: Arckpack (Empresa de Embalagens)

Endereço: www.arckpack.com.br

Elementos de Hiperímia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	0	Baixa	x 30	0	0
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	1	Baixa	x 30	30	30
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Text Unit</i>	24	Baixa	x 1	24	34
	5	Média	x 2	10	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	12	Baixa	x 2	24	24
	0	Média	x 6	0	
	0	Alta	x 12	0	
<i>Animation Unit</i>	0	Baixa	x 15	0	60
	1	Média	x 60	60	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	550				
<b>Total de pontos: 698</b>					

Site: Exatur (Agência de Viagens)

Endereço: [www.viaexatur.com.br](http://www.viaexatur.com.br)

Elementos de Hiperímia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	0	Baixa	x 30	0	0
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	1	Baixa	x 30	30	30
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Text Unit</i>	4	Baixa	x 1	4	4
	0	Média	x 2	0	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	31	Baixa	x 2	62	392
	55	Média	x 6	330	
	0	Alta	x 12	0	
<i>Animation Unit</i>	1	Baixa	x 15	15	15
	0	Média	x 60	0	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	300				
<b>Total de pontos: 741</b>					

Site: ZPTratores (Empresa especializada na venda de tratores)

Endereço: www.zptratores.com.br

Elementos de Hiperímia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	0	Baixa	x 30	0	0
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	1	Baixa	x 30	30	30
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Text Unit</i>	1	Baixa	x 1	10	10
	0	Média	x 2	0	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	73	Baixa	x 2	146	146
	0	Média	x 6	0	
	0	Alta	x 12	0	
<i>Animation Unit</i>	0	Baixa	x 15	0	0
	0	Média	x 60	0	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	300				
<b>Total de pontos: 486</b>					

Site: Raperfurados (Empresa de venda de metais perfurados)

Endereço: www.raperfurados.com.br

Elementos de Hiperímia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	0	Baixa	x 30	0	0
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	1	Baixa	x 30	30	30
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Text Unit</i>	4	Baixa	x 1	4	4
	0	Média	x 2	0	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	49	Baixa	x 2	98	332
	29	Média	x 6	174	
	5	Alta	x 12	60	
<i>Animation Unit</i>	0	Baixa	x 15	0	60
	1	Média	x 60	60	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	100				
<b>Total de pontos: 526</b>					



Site: CMI (Câmara Municipal de Itirapina)

Endereço: [www.camaraitirapina.sp.gov.br](http://www.camaraitirapina.sp.gov.br)

Elementos de Hiperímídia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	3	Baixa	x 30	90	90
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	4	Baixa	x 30	120	120
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Text Unit</i>	201	Baixa	x 1	201	201
	0	Média	x 2	0	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	8	Baixa	x 2	16	16
	0	Média	x 6	0	
	0	Alta	x 12	0	
<i>Animation Unit</i>	1	Baixa	x 15	15	15
	0	Média	x 60	0	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	550				
<b>Total de pontos: 992</b>					

Site: Tapetes-SC (Fábrica de Tapetes)

Endereço: www.tapetessaocarlos.com.br

Elementos de Hiperímia	Complexidade			Complexidade Total	Total Tipo Elemento
<i>Data Unit</i> ( <i>MultiData Unit</i> )	5	Baixa	x 30	150	150
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	2	Baixa	x 30	60	190
	1	Média	x 50	50	
	1	Alta	x 80	80	
<i>Text Unit</i>	4	Baixa	x 1	4	6
	1	Média	x 2	2	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	521	Baixa	x 2	1042	1288
	23	Média	x 6	138	
	9	Alta	x 12	108	
<i>Animation Unit</i>	0	Baixa	x 15	0	130
	0	Média	x 60	0	
	1	Alta	x 130	130	
Vetorização ( <i>Image Unit</i> )	1	Baixa	x 20	20	20
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	550				
<b>Total de pontos: 2334</b>					

Site: Landart (Loja de venda de produtos para artesanato)

Endereço: www.landart.com.br

<b>Elementos de Hiperímia</b>	<b>Complexidade</b>			<b>Complexidade Total</b>	<b>Total Tipo Elemento</b>
<i>Data Unit</i> ( <i>MultiData Unit</i> )	1	Baixa	x 30	30	30
	0	Média	x 50	0	
	0	Alta	x 80	0	
<i>Entry Unit</i>	3	Baixa	x 30	90	140
	1	Média	x 50	50	
	0	Alta	x 80	0	
<i>Text Unit</i>	1	Baixa	x 1	1	3
	1	Média	x 2	2	
	0	Alta	x 4	0	
<i>Image Unit</i> ( <i>MultiImage Unit</i> )	304	Baixa	x 2	608	704
	10	Média	x 6	60	
	3	Alta	x 12	36	
<i>Animation Unit</i>	1	Baixa	x 15	15	15
	0	Média	x 60	0	
	0	Alta	x 130	0	
Vetorização ( <i>Image Unit</i> )	0	Baixa	x 20	0	0
	0	Média	x 45	0	
	0	Alta	x 90	0	
Relação Estrutural	550				
<b>Total de pontos: 1442</b>					