

Cláudio de Miranda Luz

SOFIA – Um arcabouço para o desenvolvimento
de ferramentas de *group storytelling*

Dissertação de Mestrado apresentada ao
Programa de Pós - Graduação em
Informática, Instituto de Matemática,
Núcleo de Computação Eletrônica,
Universidade Federal do Rio de Janeiro
como requisito parcial à obtenção do
título de Mestre em Informática

Maria Luiza Machado Campos, Ph.D.
Marcos Roberto da Silva Borges, Ph.D.

Rio de Janeiro
2008

L979 Luz, Cláudio de Miranda.

Sofia: um arcabouço para o desenvolvimento de ferramentas de group storytelling / Cláudio de Miranda Luz. - Rio de Janeiro; 2008. 131f.:il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica, 2008.

Orientadora: Maria Luiza Machado Campos.

Co-Orientador: Marcos Roberto da Silva Borges.

1. Group Storytelling. - Teses. 2. Conhecimento Coletivo - Teses. 3. Gestão do Conhecimento. - Teses 4. Arcabouço. - Tese 5. Reuso – Teses. I. Maria Luiza Machado Campos (Orient.). II. Marcos Roberto da Silva Borges (Co-Orient.). III. Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica. IV. Título.

CDD

Cláudio de Miranda Luz

Sofia – Um Arcabouço para o desenvolvimento de ferramentas de *group storytelling*

Dissertação submetida ao corpo docente do Instituto de Matemática e do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do grau de Mestre em Informática.

Aprovada em: Rio de Janeiro, 24 de outubro de 2008.

Prof^a. Maria Luiza Machado Campos, Ph.D. – PPGI/UFRJ

Prof. Marcos Roberto da Silva Borges, Ph.D. – PPGI/UFRJ

Prof^a. Vannesa Braganholo Murta, D.Sc. – PPGI/UFRJ

Prof^a. Itana Maria de Souza Gimenes, Ph.D. – DIN/UEM

Agradecimentos

À Maria Luiza e ao Marcos Borges, pela orientação, apoio, compreensão, incentivo, paciência e ensinamentos transmitidos no decorrer de todo o mestrado;

Paulo Pires, Carlo, Vanessa, Amauri, José Orlando e todos os outros professores com os quais pude conviver e aprender;

Bernardo Blasquez e Henrique Silva pela presteza com que puderam me ajudar com as atividades inerentes a este trabalho;

Aos meus companheiros de trabalho da Globo.com que me apoiaram e incentivaram durante esta jornada;

Meus amigos que estavam presentes nas horas boas e ruins e que me incentivaram sempre;

Aúna, Teka e Tekinha que ficaram ao meu lado transmitindo a paz que eu precisava;

E principalmente, a meus pais e meu irmão, pela presença, pelo apoio incondicional, pela compreensão e pela paciência com que me ajudaram durante todo o mestrado.

RESUMO

LUZ, Cláudio de Miranda. **Sofia – Um arcabouço para o desenvolvimento de ferramentas de *group storytelling***. 2008. 130 f.. Dissertação (Mestrado em Informática). PPGI, Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008.

Existe um número considerável de estudos que aplicam a técnica de *group storytelling* para recuperar conhecimento tácito de um grupo de pessoas. Apesar de aplicarem esta técnica em diferentes contextos, estes estudos têm muitos pontos em comum. As ferramentas desenvolvidas para apoiar a dinâmica de *group storytelling* também têm muitas características semelhantes e, apesar disso, elas foram construídas ignorando trabalhos anteriores. Houve pouco ou nenhum reuso de software. Esta dissertação apresenta um arcabouço para ser usado no desenvolvimento de ferramentas de apoio ao *group storytelling*. O arcabouço agrega funcionalidades necessárias para qualquer solução de TI relacionada a este tema.

ABSTRACT

LUZ, Cláudio de Miranda. **Sofia – Um arcabouço para o desenvolvimento de ferramentas de *group storytelling***. 2008. 130 f.. Dissertação (Mestrado em Informática). PPGI, Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008.

There has been a considerable number of studies applying the technique of group storytelling to collect tacit knowledge from people in a group. In spite of applying it on different contexts, these studies have many similarities on how this is done. The tools developed to support the group storytelling tasks also have many characteristics in common and, despite that, they were constructed ignoring many previous works. There has been little or no reuse. This dissertation presents a framework to be applied in the development of group storytelling supporting tools. The framework encompass functionalities necessary to every IT solution related to this subject.

LISTA DE FIGURAS

Figura 1 - Espiral do conhecimento.....	24
Figura 2 - Análise das etapas da dinâmica de <i>group storytelling</i> dos trabalhos estudados	35
Figura 3 - Estrutura de uma história segundo Perret (2004)	39
Figura 4 - Estrutura de uma história segundo Oliveira (2006).....	40
Figura 5 - Estrutura de uma história segundo Carminatti (2006).....	42
Figura 6 - Estrutura de uma história segundo Laporti (2007)	42
Figura 7 - Estrutura de uma história segundo Reis (2007).....	43
Figura 8 - Comparação entre design OO tradicional e processo de desenvolvimento de arcabouços - extraído de Markiewicz e Lucena (2007).....	53
Figura 9 - Etapas do Projeto Dirigido por Exemplo para o desenvolvimento de um arcabouço - extraída de (SILVA, 2000).....	54
Figura 10 - As etapas do Projeto Dirigido Por Hot Spot para o desenvolvimento de um arcabouço - extraída de (SILVA, 2000).....	55
Figura 11 - Estrutura de uma história no arcabouço.....	66
Figura 12 - Modularização baseada no modelo 3C	70
Figura 13 - Visão global da solução	79
Figura 14 - Diagrama representando o padrão Model 2 - extraída de (VIADIGITAL, 2007). 82	
Figura 15 - Diagrama de seqüência da funcionalidade Cadastrar História.....	88
Figura 16 - Modelo de dados do arcabouço	89
Figura 17 - Ambiente de instanciação do arcabouço.....	92
Figura 18 - Fluxograma de decisão para instanciação do arcabouço	95
Figura 19 - Relatório de métricas do arcabouço puro.....	98
Figura 20 - Relatório de métricas da instanciação do arcabouço	99
Figura 21 - Modelo de dados completo	128

LISTA DE TABELAS

Tabela 1 - Distinção intuitiva entre informação e conhecimento - extraída de Zeleny (2005)	21
Tabela 2 - Papéis presentes em cada um dos trabalhos de <i>group storytelling</i>	37
Tabela 3 - Quadro comparativo geral - parte I	46
Tabela 4 - Quadro comparativo geral - parte II	47
Tabela 5 - Taxonomia tempo x espaço para o elemento comunicação	68
Tabela 6 - Quadro com níveis de flexibilidade oferecidas pelo arcabouço	73
Tabela 7 - <i>Templates</i> pré-estabelecidos para configuração de histórias	74
Tabela 8 - Estrutura de pacotes do arcabouço de acordo com o modelo 3C	85
Tabela 9 - Controle de transação com AOP	86
Tabela 10 - Relação entre funcionalidades e entidades do modelo de dados	89
Tabela 11 - Quadro com métricas e aumento percentual	99

LISTA DE ABREVIATURAS

AOP – *Aspect-oriented programming*
CSCW – *Computer Supported Cooperative Work*
DBC – Desenvolvimento baseado em Componentes
GC – Gestão do Conhecimento
GST – *Group StoryTelling*
OO – Orientado a objetos
POJO – *Plain Old Java Objects*
SGBD – Sistema gerenciador de banco de dados
TI – Tecnologia da Informação

Sumário

Agradecimentos	4
LISTA DE FIGURAS	7
LISTA DE TABELAS	8
LISTA DE ABREVIATURAS	9
Sumário	10
1 Introdução	12
1.1 Motivação	12
1.2 O problema	13
1.3 Enfoque de solução	15
1.4 Hipótese e objetivos	17
1.5 Organização do trabalho	18
2 Gestão do conhecimento e <i>group storytelling</i>	20
2.1 O Conhecimento	20
2.1.1 Dado, informação e conhecimento	20
2.1.2 Conhecimento tácito e conhecimento explícito	22
2.1.3 Gestão do conhecimento	25
2.2 Recuperação coletiva do conhecimento e <i>group storytelling</i>	27
2.2.1 O trabalho cooperativo	27
2.2.2 Recuperação coletiva do conhecimento	28
2.2.3 <i>Group storytelling</i>	29
3 Soluções de apoio ao <i>group storytelling</i>	31
3.1 Soluções de <i>group storytelling</i> estudadas	31
3.2 Fatores de comparação	32
3.2.1 Etapas da dinâmica	33
3.2.2 Papéis envolvidos	35
3.2.3 Estruturação da história	37
3.2.4 Insumos e produtos gerados	44
3.2.5 Quadro comparativo geral	45
3.3 Outras soluções	48
3.4 Problemas com as constantes re-implementações	49
4 Arcabouços	51
4.1 Arcabouço como solução	51
4.2 Metodologias de desenvolvimento de arcabouços	52
4.3 Classificação de arcabouços	56
4.3.1 Quanto à técnica de utilização	57
4.3.2 Quanto ao escopo de utilização	57
4.4 Ciclo de vida de um arcabouço	58
5 Sofia - Concepção	60
5.1 Um arcabouço como proposta de solução	60
5.2 Abordagem deste trabalho	61
5.3 Características herdadas das soluções estudadas	63
5.3.1 Etapas da dinâmica	63
5.3.2 Papéis envolvidos	64
5.3.3 Estruturação da história	65
5.3.4 Insumos e produtos	67
5.4 Funcionalidades presentes no arcabouço classificadas segundo o modelo 3C	67

5.4.1	<i>Groupware</i> baseado no modelo 3C	67
5.4.2	Funcionalidades	70
5.5	Flexibilidade e variabilidade oferecidas pelo arcabouço.....	72
5.6	Classificação do arcabouço desta dissertação.....	75
6	Sofia - Implementação	77
6.1	Alicerces de construção do arcabouço	77
6.2	Visão geral do arcabouço	78
6.2	Linguagem de programação	79
6.3	Padrões	80
6.3.1	Model 2 como macro arquitetura	81
6.3.2	Padrões de projeto	82
6.4	Detalhes de implementação da interface	83
6.5	Componentes	84
6.5.1	Controle de transação com AOP	86
6.6	<i>Mashups</i>	86
6.7	Interação entre as classes	88
6.8	Modelo de dados	88
6.9	Ambiente de instanciação.....	91
6.10	Passo-a-passo para instanciação do arcabouço.....	92
7	Avaliação do arcabouço Sofia	96
7.1	Abordagem de avaliação	96
7.2	Primeira abordagem	97
7.3	Segunda abordagem	101
7.3.1	Relatório.....	101
7.3.2	Considerações sobre o relatório de <i>feedback</i>	103
7.4	Considerações gerais	104
8	Conclusão	106
8.1	Considerações finais.....	106
	Referências	110
	Anexo 1 – Preparação do ambiente	116
A1	Instalação dos sistemas de gerência da base de dados	116
A1.1	Instalação do MySQL	116
A1.2	Instalação das ferramentas gráficas para manipulação do banco.....	116
A2	Criação e preparação da base de dados	117
A2.1	Criar o Schema	117
A3	Instalação do servidor de aplicações	118
A3.1	Instalação do JBoss.....	118
A4	Configuração dos serviços no servidor de aplicações	118
A4.1	Configurando o datasource	118
A4.2	Configurando o serviço de e-mail	118
A4.3	Habilitando AOP no JBoss	119
A5	Implantação e teste do arcabouço	120
Script de criação das tabelas (create_sofia.sql)		120
Script de carga inicial do banco (insert_sofia.sql).....		125
Arquivo de configuração do datasource (sofia-mysql-ds.xml).....		126
Arquivo de configuração do serviço de e-mail (sofia-mail-service.xml).....		127
	Anexo 2 – Modelo de dados.....	128
Descrição de cada uma das entidades		129

1 Introdução

1.1 Motivação

Na segunda metade do século XX, principalmente no final deste período, as organizações começaram a perceber que seu maior ativo era o conhecimento. A sobrevivência dentro do cenário social e econômico ia depender do quanto elas investiriam na gestão deste ativo. O elemento conhecimento ganhou relevância e isso se refletiu diretamente na vida das organizações. Segundo Drucker *apud* Schreiber *et al.* (1999, p.2), “Conhecimento é o único recurso significativo atualmente. Os tradicionais ‘fatores de produção’ – terra (recursos naturais), mão-de-obra e capital – não desapareceram. Mas eles se tornaram secundários. Eles podem ser obtidos, e obtidos facilmente, desde que exista conhecimento”.

Os produtos, serviços e atividades em geral cada vez mais tinham seu valor representado pelo conhecimento agregado e não pelos elementos tangíveis relacionados a eles. Esta importância percebida fez surgir a disciplina chamada de gestão do conhecimento. Essa disciplina agrega atividades como identificação, aquisição, desenvolvimento, disseminação, utilização e preservação do conhecimento dentro da organização.

O conhecimento ao qual nos referimos não é somente o explícito, aquele de fácil manipulação, presente em relatórios, documentos, tabelas e outros artefatos, mas principalmente o tácito, que é intangível, que está presente na mente das pessoas, e cuja externalização é complexa. A distinção entre conhecimento tácito e explícito foi estabelecida por Michael Polanyi (1966) e é muito utilizada na área de gestão do conhecimento.

Um dos grandes desafios da área de gestão de conhecimento se refere à captura e disseminação do conhecimento tácito. Práticas, procedimentos e ferramentas são estudados, propostos e utilizados com o intuito de promover o compartilhamento do conhecimento tácito que existe nas organizações. Esse conhecimento está presente em cada um dos indivíduos que

atuam dentro de uma organização, entretanto, com a natureza cada vez mais colaborativa das atividades executadas por eles atualmente, a gestão do conhecimento deve considerar não somente o conhecimento tácito de um indivíduo, e sim de um grupo.

1.2 O problema

A relevância adquirida pelo elemento conhecimento na segunda metade do século passado está intrinsecamente ligada ao advento da computação eletrônica. O uso de computadores pessoais se difundiu em praticamente todas as organizações ao redor do globo. Usando computadores as pessoas podem fazer o que faziam antes de maneira mais rápida, mais barata, mais segura e com menos esforço. Assim, os trabalhadores se tornaram mais eficientes. Praticamente todos os processos existentes dentro das organizações passaram a ser executados com o apoio computacional, e novos processos surgiram com essa mudança.

Na última década do século XX, com a popularização da Internet, outra grande barreira foi ultrapassada. A questão da distância deixou de ser um problema, a comunicação entre as pessoas foi facilitada, o acesso à informação cresceu conforme a grande rede se enriqueceu de conteúdo, e a Internet virou uma infra-estrutura quase que essencial para uma boa parte das organizações.

A mesma tecnologia que trouxe facilidades para o trabalho dentro das organizações, que possibilitou uma nova abordagem de trabalho colaborativo, também impôs novos desafios para a área de gestão do conhecimento. A troca de informações, a comunicação, a maneira de encarar os problemas e suas soluções, decisões tomadas, tudo que faz parte da memória organizacional, que ajuda a organização no futuro, passou a ser realizado de maneira diferente. Por conseguinte, o modo de se fazer gestão do conhecimento teve que acompanhar essa evolução.

Com as pessoas cada vez mais trabalhando com o apoio computacional, usando meios de comunicação como e-mail e mensagens instantâneas, fóruns de discussão, blogs

corporativos, dentre outros meios, é natural que o conhecimento tácito fique armazenado nos registros dessas aplicações. Mesmo que seja de forma não estruturada, o conhecimento está presente ali; decisões tomadas podem estar contidas numa troca de e-mails, opções de projeto podem estar registradas no histórico de mensagens instantâneas, experiências ruins podem ser encontradas em fóruns de discussão, novas tendências podem estar sendo discutidas em blogs. Também é comum que sejam feitos relatórios para registro das boas práticas e das experiências ruins, mas muitas vezes boa parte do conhecimento tácito fica omitido em todos esses meios de armazenamento.

Para estimular a externalização do conhecimento tácito, as pessoas da área de gestão do conhecimento começaram a estudar e propor práticas para esse fim. Uma dessas práticas é a técnica de *group storytelling*. Os membros de uma equipe, distribuídos ou no mesmo local, contribuem para criar uma história em conjunto, de forma assíncrona ou síncrona, utilizando diferentes mídias (VALLE; PRINZ; BORGES, 2002). O intuito dessa técnica é juntar numa única história os elementos tácitos do conhecimento de cada um dos participantes que contribuíram para a construção da mesma.

O ato de contar uma história não necessariamente exige o uso de um apoio computacional, mas existem diversos requisitos que fazem com que o uso do computador seja de extrema valia para o *group storytelling*. A possibilidade de se trabalhar de forma distribuída é uma delas. Muitas vezes é difícil reunir as pessoas num mesmo local, e isso ainda é mais evidente quando esse grupo é grande. Outra vantagem é permitir que as pessoas contribuam de forma assíncrona. Essa flexibilidade estimula e aumenta as chances de que os participantes contribuam da melhor maneira possível. Além disso, ao se usar o computador, o processo de registro das contribuições e do histórico de construção da história é automático. As histórias construídas coletivamente são como um grafo onde os nós são pequenos trechos

de texto e as ligações entre os nós são associações que têm algum significado (LUZ; BORGES; CAMPOS, 2008).

Alguns trabalhos já adotaram o uso dessa técnica e propuseram soluções de TI para apoio. Perret (2004) implementou a ferramenta Tellstory que serviu de guia para alguns trabalhos posteriores. Oliveira (2006) usou a Tellstory como base e implementou a ferramenta Feedback para captura de histórias coletivas de uso de sistemas. Carminatti (2006) fez adaptações ao Tellstory e utilizou essa solução em seu trabalho relacionado a situações de emergência. Silva (2006) implementou uma solução baseada em *group storytelling* para resolução de conflitos. Outras pesquisas, como Laporti (2007) e Reis (2007), foram concluídas recentemente e desenvolvem soluções específicas para seus problemas.

Normalmente as soluções de TI que são adotadas para apoio a algum processo sofrem uma evolução com o tempo. Muitas vezes elas são adaptadas ou totalmente reimplementadas para cada caso, mesmo que eles tenham várias características em comum. Isso aconteceu com as implementações para apoio ao *group storytelling*, então o problema que vamos considerar como guia para esse trabalho é:

Como possibilitar que soluções baseadas em TI para apoio ao *group storytelling* sejam desenvolvidas a partir de uma base já estabelecida aproveitando alicerces definidos a priori?

1.3 Enfoque de solução

A capacidade de raciocinar do ser humano faz com que ele se diferencie dos outros animais da natureza. O Homem é capaz de aprender com os erros que comete e também com os acertos. Quando uma solução resolve um problema, ela passa a ser aplicada sempre que esse problema reaparece. Essa associação problema-solução vale tanto para pequenas

dificuldades do dia-a-dia como para enormes desafios científicos, não importando a área da ciência que estejamos considerando. Arquitetura, economia, medicina, ou engenharia de software que é o nosso caso, qualquer área de conhecimento humano tem os seus pares problema-solução.

Na área de computação, mais especificamente na área de software, o reaproveitamento de uma solução aplicada a um problema é chamado de reuso de software. Sub-rotinas nos anos 60, módulos nos anos 70, objetos nos anos 80 e componentes nos anos 90 (ALMEIDA, 2007, p.14), os engenheiros de software também têm os seus catálogos de pares problema-solução e com a velocidade com que essa ciência avança, esse catálogo cresce, evolui, e se renova a cada dia.

O reuso se aplica não só à solução concreta, mas também à solução conceitual. Os padrões de projeto, ou *design patterns* como são mais conhecidos, são soluções conceituais. Podem ser definidos de forma resumida como soluções comuns para problemas recorrentes, pois são reusados sempre que aquele mesmo problema aparece. Uma solução que resolve um problema num determinado contexto e que é consenso dentre a maioria passa a ser um padrão de projeto. A vantagem de ser uma solução conceitual é que ela pode ser instanciada no contexto em que o problema acontece.

Objetos, componentes, e padrões de projeto, podem ser considerados elementos de reuso para problemas de escopo simples, ou de menor escopo. A partir do momento que esses elementos passam a ser combinados, e combinados da mesma maneira repetidas vezes, para resolver um problema de escopo maior, podemos considerar que um novo par problema-solução está se formando. Normalmente soluções de maior escopo como essa são denominadas de arcabouços ou no termo em inglês, *frameworks*.

1.4 Hipótese e objetivos

A constante re-implementação de soluções de TI para apoio à dinâmica de *group storytelling* representa um desgaste desnecessário de esforço e tempo que poderiam estar sendo empregados de outro modo. Uma maneira de evitar esse desperdício é partir de um patamar onde os alicerces de construção já tenham sido estabelecidos e apenas algumas variações tenham que ser implementadas. Logo, podemos supor que a utilização de um arcabouço como plataforma inicial de desenvolvimento destes sistemas exercerá exatamente este papel.

A proposta deste trabalho é definir e implementar um arcabouço que atenda as expectativas das pessoas que pretendem desenvolver sistemas de apoio ao *group storytelling*. O arcabouço deverá ser flexível o bastante para permitir a instanciação de acordo com os requisitos funcionais e não funcionais da situação em questão e deverá fazer com que a construção destes sistemas seja feita de forma mais rápida. Se for possível avaliarmos que a utilização do arcabouço evitou o re-trabalho e poupou esforço então o arcabouço terá exercido seu papel.

A hipótese de solução para o problema descrito neste capítulo pode ser registrada formalmente da seguinte maneira:

É possível se definir um arcabouço como base para o desenvolvimento de soluções de apoio ao *group storytelling* de modo que a implementação de um sistema comece de um patamar já com alicerces definidos.

O objetivo deste trabalho, portanto, é desenvolver este arcabouço detalhando os alicerces de construção juntamente com um passo-a-passo de instanciação, e mostrar, no final,

que o desenvolvimento de sistemas de apoio ao *group storytelling* a partir do arcabouço exige muito menos esforço por parte dos desenvolvedores do que se eles tivessem começado uma implementação sem nenhuma plataforma inicial.

1.5 Organização do trabalho

No capítulo 2, são apresentados os conceitos ligados à área gestão do conhecimento, com detalhamento sobre a hierarquia dado, informação e conhecimento e sobre a diferenciação entre conhecimento tácito e conhecimento explícito. Além disso, também falamos sobre recuperação coletiva do conhecimento com ênfase na técnica de *group storytelling*.

No capítulo 3, fazemos uma análise sistemática de algumas soluções de apoio ao *group storytelling* observando aspectos como: etapas da dinâmica, papéis envolvidos, estruturação da história e insumos e produtos gerados. Além disso, fazemos uma breve passagem sobre algumas outras soluções de TI relacionadas a *storytelling*. Fechamos o capítulo relacionando os problemas encontrados com as constantes re-implementações.

No capítulo 4, falamos especificamente sobre arcabouços apresentando definições e estudando algumas metodologias de desenvolvimento. Depois são apresentados alguns tipos de classificação que ajudam a compreender este artefato e também é discutida a questão do ciclo de vida de um arcabouço.

No capítulo 5, é apresentado o arcabouço que é a proposta desta dissertação observando-se inicialmente as características herdadas das soluções estudadas e em seguida os alicerces de construção que caracterizam a proposta. No final do capítulo o arcabouço é classificado conforme os critérios apresentados no capítulo 4.

No capítulo 6 são apresentados os detalhes concretos de implementação onde falamos sobre linguagem de programação, padrões de projeto adotados, componentes utilizados,

tecnologias selecionadas, dentre outras questões técnicas. Além disso, são apresentadas algumas funcionalidades pré-implementadas e o passo-a-passo de instanciação do arcabouço.

No capítulo 7 são apresentados os detalhes da avaliação que foi feita com o arcabouço descrevendo os critérios e os principais resultados que foram obtidos.

Enfim, no capítulo 8, são discutidas as conclusões, contribuições e limitações do arcabouço proposto nesta dissertação, além de possíveis trabalhos futuros que podem ser desenvolvidos a partir dele.

2 Gestão do conhecimento e *group storytelling*

Neste capítulo apresentaremos os conceitos ligados à área de gestão do conhecimento iniciando com definições sobre o que é conhecimento e apresentando alguns tipos de classificação. Em seguida vamos falar sobre conhecimento coletivo fazendo um paralelo com o trabalho cooperativo nas organizações e o uso de sistemas de apoio a essas atividades. Finalizaremos falando especificamente sobre *group storytelling*.

2.1 O Conhecimento

2.1.1 Dado, informação e conhecimento

Para entendermos melhor o significado da palavra conhecimento dentro do contexto deste trabalho, vamos analisar também os termos dado e informação verificando qual é a relação e as diferenças entre eles. Normalmente dado, informação e conhecimento são apresentados numa hierarquia que alguns autores ainda estendem com os termos sabedoria e ‘*enlightenment*’. Essa hierarquia aparece na literatura tanto no domínio de gestão do conhecimento como no de ciência da informação. Sharma (2008) fez um estudo da origem da mesma.

De acordo com a hierarquia, os dados seriam os elementos mais simples, que quando isolados teriam pouca relevância ou valor. Eles são apenas um conjunto de fatos distintos e objetivos relativos a eventos. Sua importância aparece quando percebemos que ele é essencial para a criação da informação.

Informação é dado dotado de relevância e propósito. Informação pode ser entendida como dado que faz alguma diferença, que causa algum impacto na pessoa que a recebe. Segundo Davenport e Pruzak (1998), dados se transformam em informação quando se

adiciona valor a eles por meio de métodos como: contextualização, categorização, cálculo, correção e condensação.

O conhecimento normalmente é visto como algo bem mais amplo do que dado ou informação. O significado do termo conhecimento dentro das organizações é o que nos interessa neste trabalho e é apresentado por Davenport e Pruzak (1998) como:

uma mistura fluida de experiência condensada, valores, informação contextual e insight experimentado, a qual proporciona uma estrutura para a avaliação e incorporação de novas experiências e informações. Ele tem origem e é aplicado na mente dos conhecedores. Nas organizações, ele costuma estar embutido não só em documentos ou repositórios, mas também em rotinas, processos, práticas e normas organizacionais (DAVENPORT; PRUZAK, 1998. p.5).

Por essa definição podemos perceber que conhecimento é algo bem mais complexo do que dado ou informação. Nós dizemos que uma pessoa conhece ou possui determinado conhecimento quando acreditamos que essa pessoa seja capaz de tomar decisões e agir com foco em algum objetivo. Zeleny (2005) apresenta um quadro que nos ajuda a distinguir de forma intuitiva a diferença entre informação e conhecimento.

Tabela 1 - Distinção intuitiva entre informação e conhecimento - extraída de Zeleny (2005)

Informação	Conhecimento
Pode ser demais	Nunca é suficiente
É uma coisa	É um processo
Alguém pode possuí-la	Alguém deve demonstrá-lo
Pedaço por pedaço	Sempre um todo
Certa ou errada	Mais ou menos
Individualmente confirmada	Socialmente aprovado

Com o advento da Internet e com o fácil acesso das pessoas a todo tipo de conteúdo nós passamos a ouvir a expressão “sobrecarga de informação”. Entretanto, não se houve falar em “sobrecarga de conhecimento”. Conhecimento nunca é demais.

Davenport e Pruzak (1998) entendem que assim como informações são derivadas de dados, o conhecimento deriva da informação, e isso se dá através de quatro processos que necessariamente envolvem pessoas:

- Comparação: como a informação sobre uma situação se compara a outras situações que conhecemos?
- Conseqüências: que implicações a informação traz sobre decisões e ações?
- Conexões: Como esse pedaço de conhecimento se relaciona com outros?
- Conversação: O que as outras pessoas pensam sobre essa informação?

Na hierarquia que estamos considerando, além de dado, informação e conhecimento, ainda encontramos na literatura o elemento sabedoria, e mais recentemente, o termo ‘*enlightenment*’ no topo. Para os objetivos deste trabalho podemos considerar esses elementos todos como conhecimento, mas para uma breve explicação podemos recorrer à metáfora usada por Zeleny, que associa dado, informação, conhecimento e sabedoria com “saber-nada”, “saber-oque”, “saber-como” e “saber-porque”, respectivamente. Segundo Zeleny, *apud* Sharma (2008, p.2), o nível de ‘*enlightenment*’ seria “não apenas responder ou entender o porquê (sabedoria), mas alcançar o senso da verdade absoluta, o senso do certo e do errado, e tê-lo socialmente aceito, respeitado e sancionado”.

2.1.2 Conhecimento tácito e conhecimento explícito

Uma classificação comumente utilizada em gestão do conhecimento é aquela que distingue o conhecimento entre tácito e explícito. Essa classificação foi definida por Polanyi (1966) e, muitas vezes, orienta a maneira como as organizações lidam com a captura e a disseminação do conhecimento.

O conhecimento explícito é de fácil compartilhamento, estando ligado a coisas mais objetivas e concretas. Pode ser representado por fórmulas, especificações, documentos, textos, relatórios, tabelas ou outros artefatos similares. A objetividade do conhecimento explícito o torna mais fácil de se articular e de ser expresso em termos claros.

Em contrapartida, o conhecimento tácito está associado à subjetividade. Ele é difícil de ser formalizado e, portanto, não é fácil compartilhá-lo. Segundo Nonaka (1998), ele está ligado a habilidades técnicas: o tipo de destreza informal e de difícil especificação, incorporada ao termo know-how e também abrange modelos mentais, crenças e perspectivas tão arraigadas que são tidas como algo certo, não sujeitas a fácil manifestação.

Quando Davenport e Pruzak, em sua definição de conhecimento, mencionam “mistura fluida de experiência condensada, valores, informação contextual e insight experimentado” podemos considerar que eles estão falando de conhecimento tácito enquanto que quando citam “documentos”, “repositórios” e “rotinas, processos, práticas e normas organizacionais” eles estão falando de conhecimento explícito.

A associação dois a dois destas duas categorias de conhecimento geram quatro modos de conversão do conhecimento, que são apresentados por Nonaka e Takeuchi (1997) como:

- **Socialização** (tácito para tácito): consiste na transmissão direta de conhecimento tácito. Ocorre, em geral, através da observação, imitação e prática. São compartilhados modelos mentais, pontos de vista, experiências;
- **Combinação** (explícito para explícito): ocorre quando há combinação de vários conhecimentos explícitos para a formação de um novo todo;

- **Externalização** (tácito para explícito): padrão em que novo conhecimento explícito é gerado através da formalização de um conhecimento tácito apresentado de forma categorizada e contextualizada.
- **Internalização** (explícito para tácito): consiste na utilização de conhecimentos explícitos para a ampliação, extensão, e reformulação do conhecimento tácito dos indivíduos.

A interação contínua e dinâmica entre o conhecimento tácito e o explícito, moldada pelos quatro processos de conversão do conhecimento, formam o que Nonaka e Takeuchi chamam de espiral do conhecimento (Figura 1). A espiral é um ciclo de vida que está presente nas organizações criadoras de conhecimento.



Figura 1 - Espiral do conhecimento

O conceito de conhecimento explícito não é adotado por alguns autores na literatura. Zeleny (2006) considera essa categoria de conhecimento como informação, e conhecimento tácito seria apenas conhecimento. Equivalentemente a Nonaka e Takeuchi ele cita quatro modos de transformação: articulação (conhecimento para informação), combinação (informação para informação), internalização (informação para conhecimento) e socialização (conhecimento para conhecimento).

Independente da maneira como se classifica o conhecimento, o que vai fazer diferença para a organização é o modo como ela trata deste ativo e de que maneira ela promove os quatro modos de conversão que fazem parte da espiral. Tudo isso ajuda a construir o domínio de gestão do conhecimento.

2.1.3 Gestão do conhecimento

A percepção por parte das organizações de que era necessário se gerenciar o capital intelectual que elas possuíam fez surgir a área de gestão do conhecimento. Métodos, procedimentos, teorias e ferramentas começaram a aparecer com o objetivo de promover o ciclo de vida do conhecimento.

Uma análise das diferentes abordagens de se fazer gestão do conhecimento permite que se possa identificar atividades comuns entre elas. Schreiber *et al.* (1999) fez essa análise e enumerou uma série de atividades que são apresentadas na forma de uma cadeia de valor do conhecimento. São elas:

- **Identificar** conhecimento existente interna e externamente;
- **Planejar** qual conhecimento será necessário no futuro;
- **Adquirir e/ou desenvolver** o conhecimento necessário;
- **Distribuir** o conhecimento para onde ele é necessário;

- **Encorajar** a aplicação do conhecimento nos processos de negócio da organização;
- **Controlar e manter** a qualidade do conhecimento;
- **Descartar** o conhecimento quando ele não for mais necessário.

Ainda segundo Schreiber *et al.*, o objetivo da área de GC é “melhorar a infra-estrutura de conhecimento de uma organização, com o objetivo de levar o conhecimento certo para as pessoas certas, no formato certo, no momento certo”. As atividades da cadeia estão definidas para buscar esse objetivo.

A definição apresentada por Schreiber *et al.* é simples e prática e conta com a intuição que temos sobre o assunto. Uma definição mais elaborada é aquela definida por Terra e Gordon (2002) *apud* Perret (2004):

a GC, em seu sentido mais atual, pode ser considerada o esforço para melhorar o desempenho humano e organizacional por meio da facilitação de conexões significativas. Em termos práticos, isso significa: garantir que todos dentro da organização tenham acesso ao conhecimento da organização, quando, onde e na forma que eles necessitam; ajudar e motivar que detentores de conhecimentos importantes compartilhem seu conhecimento, tornando mais simples o processo para esses indivíduos codificarem parte de seu conhecimento e/ou colaborarem com outros (TERRA; GORDON, 2002).

Em geral, o objetivo da GC é a disseminação do conhecimento dentro da organização o que implica na necessidade da captura e do armazenamento deste conhecimento. Neste trabalho consideramos a definição proposta por Perret (2004) como a que resume da melhor maneira a área de GC conforme nosso foco:

Gestão do conhecimento é a disciplina que administra o conjunto de procedimentos, práticas e ferramentas que visam capturar, armazenar e disseminar o conhecimento dentre os funcionários da organização, aproveitando os recursos tecnológicos existentes (PERRET, 2004. p.28).

Além de destacar as atividades de captura, armazenamento e disseminação, essa definição encara o conhecimento como algo distribuído dentro de um grupo numa organização e deixa claro que a tecnologia pode ser uma aliada na abordagem de GC.

2.2 Recuperação coletiva do conhecimento e *group storytelling*

2.2.1 O trabalho cooperativo

A característica de trabalho cooperativo que existe dentro das organizações recebeu um impulso extraordinário após a popularização da Internet. As empresas passaram a usar a Web para facilitar a interação entre seus funcionários e estimularam que cada vez mais fossem utilizados recursos da grande rede. A barreira da distância desapareceu e a comunicação passou a ser algo trivial.

Organizações passaram a encarar desafios que exigiam conhecimentos de diferentes áreas e cada vez mais elas procuraram formar grupos de trabalho multidisciplinares. As diferentes bagagens culturais dos indivíduos asseguram uma gama variada de conhecimentos que é fundamental para a inovação (NONAKA; TAKEUCHI, 1997). Esse mosaico de pensamentos e maneiras de abordar problemas enriquece as discussões e as decisões que têm que ser tomadas no dia-a-dia das organizações.

Todavia, o trabalho em equipe carrega consigo alguns problemas que são intrínsecos a tudo que envolve interação entre pessoas. Em geral, esses problemas estão associados a questões sociais ou culturais: problemas de relacionamento interpessoal, dificuldade de comunicação, ambiente inapropriado para interação, falta de iniciativa, dentre outros fatores.

Com o crescimento do trabalho em equipe, ganhou relevância a disciplina que estudava a aplicação da tecnologia ao trabalho cooperativo, o CSCW, ou Suporte Computacional ao Trabalho Cooperativo (*Computer-Supported Cooperative Work*). Essa área

de estudo não aborda apenas as questões técnicas, mas também os aspectos sociais, culturais e psicológicos relacionados. Além da sigla CSCW, um termo que se popularizou na mesma época foi o *groupware*. Ele é usado para definir as ferramentas computacionais que são utilizadas para apoio ao trabalho cooperativo.

Dado esse cenário de trabalho cooperativo dentro das organizações, seria natural que a área de GC passasse a encarar de maneira diferente as atividades do ciclo de vida do conhecimento. A natureza coletiva das atividades desenvolvidas nas organizações implica numa distribuição do conhecimento necessário para que elas sejam executadas com sucesso. Cada pessoa contribui com suas próprias experiências e *know-how* e esse poder conjunto torna possível a realização das tarefas.

Essa característica, por sua vez, implica que para que seja feita a captura, o armazenamento e a disseminação do conhecimento, cada pessoa desse grupo dê a sua contribuição. É a união dos conhecimentos distribuídos que vai formar o capital intelectual da organização.

2.2.2 Recuperação coletiva do conhecimento

A definição de GC que estamos considerando engloba as atividades de captura, armazenamento e disseminação do conhecimento. Todas essas atividades definem um escopo muito amplo de estudo e o foco que estamos querendo adotar neste capítulo é relativo apenas à parte de captura, ou de recuperação do conhecimento tácito presente na mente das pessoas envolvidas numa tarefa coletiva.

A recuperação deste conhecimento pode ser feita através de depoimentos ou relatos. Pessoas envolvidas em determinado acontecimento podem tentar reconstruir a história relembrando detalhes, decisões e fatos que juntos formam uma versão do que aconteceu. Esta seria a versão relatada.

O relato de um episódio pode ter quatro versões: a versão guardada na mente das pessoas que testemunharam ou participaram de todos ou alguns eventos; a versão relatada por estas pessoas, isto é, a externalização de seu conhecimento tácito; a versão conhecida por estas pessoas, isto é, o conhecimento que os participantes possuem; e a descrição real ou verdadeira dos eventos, esta última provavelmente não existente (CARMINATTI; BORGES; GOMES, 2006).

Dentre estas quatro versões, a versão relatada é aquela que podemos considerar como concreta e a que podemos manipular e tentar melhorar. As outras versões são conceituais e servem para balizar a construção da versão relatada. Essa construção é feita por meio do processo de externalização, que Nonaka e Takeuchi (1997) definem como o processo de conversão de conhecimento tácito em conhecimento explícito, ou formal.

Diversos motivos fazem com que a versão relatada seja diferente da versão conhecida. As pessoas podem simplesmente esquecer de determinados fatos ou mesmo julgá-los erroneamente como irrelevantes e dessa maneira eles ficam omitidos. O conhecimento de uma pessoa é algo tão complexo que muitas vezes a percepção de seus detalhes é algo sutil. Segundo Hugh Cottan *apud* Schreiber *et al.* (1999, p.70): “Você pode saber mais do que você pensa!”. A externalização de conhecimento tácito não é trivial e depende bastante da atitude positiva e colaborativa das pessoas envolvidas.

2.2.3 Group storytelling

Quando falamos de recuperação do conhecimento baseada em relatos, podemos dizer que essa é uma técnica de construção de uma história em partes. A versão relatada é uma história de um acontecimento passado. O ato de contar a história pode ser feito individualmente com os relatos sendo lembrados por apenas uma pessoa ou em grupo com a interação entre os participantes.

A técnica de storytelling é um, dentre diversos métodos de eliciação do conhecimento. A pessoa, através de uma narrativa, descreve um acontecimento, geralmente seguindo uma ordem cronológica. Essa narrativa pode ser feita de forma verbal, e dessa maneira apenas as pessoas ao redor estarão adquirindo aquele conhecimento, ou então de forma escrita, com a possível difusão daquele conhecimento dentre um número maior de pessoas.

Uma outra abordagem para a técnica de storytelling é aquela feita em grupo, onde a história é contada de forma cooperativa e o resultado final se dá através da interação entre os participantes. A essa prática damos o nome de *group storytelling*. Diversas pessoas trabalhando de forma cooperativa produzem mais conteúdo, relembram melhor os fatos e interagem para garantir um resultado final mais enriquecido.

A dinâmica de *group storytelling* visa à recuperação do conhecimento que está contido na cabeça das pessoas. Existem diversas variações para essa dinâmica, mas o propósito final é construir uma história que seja uma representação da visão do grupo sobre determinado acontecimento. Mais de uma pessoa contribui, síncrona ou assincronamente, localmente ou de maneira distribuída, em vários pontos do processo, através de diversas mídias (VALLE; PRINZ; BORGES, 2002, p. 363).

O trabalho de retrospectiva e de eliciação do conhecimento tácito feito com grupos pode ser bem mais proveitoso do que aquele conduzido com indivíduos de forma isolada. Ao interagir em grupo é gerada uma sinergia entre os participantes, onde a contribuição de um motiva a contribuição de outro e faz com que o grupo se lembre de fatos que antes consideravam irrelevantes (CARMINATTI, 2006).

3 Soluções de apoio ao *group storytelling*

Neste capítulo apresentaremos as soluções de *group storytelling* que foram estudadas e faremos uma análise de cada uma delas sob diferentes aspectos. Em seguida abordaremos algumas outras soluções e finalizaremos discutindo os problemas considerados.

3.1 Soluções de *group storytelling* estudadas

Os estudos foram baseados em cinco diferentes trabalhos que abordaram de alguma maneira iniciativas associadas ao *group storytelling*. Todos esses trabalhos já estão concluídos e alguns foram finalizados recentemente. Discutiremos cada um seguindo a ordem cronológica com que foram propostos e desenvolvidos.

O primeiro deles foi o trabalho de Perret (2004). Este autor faz um estudo da teoria sobre o conhecimento e em seguida sobre histórias na literatura, no jornalismo e principalmente, nas organizações. Ele propõe que o *group storytelling* seja usado para elicitação de conhecimento tácito nas organizações em geral. Ele também analisa ferramentas de apoio computacional ao trabalho cooperativo citando alguns exemplos de *groupware* e detalha a dinâmica de *group storytelling* implementando a ferramenta Tellstory, que apóia a construção colaborativa de histórias. Os trabalhos que surgiram em seguida tiveram forte influência da pesquisa feita por este autor.

Após a pesquisa de Perret surgiram outros trabalhos que procuraram aplicar a técnica de *group storytelling* em situações específicas. Oliveira (2006) faz um estudo sobre gestão do conhecimento voltada para manutenção de software. A idéia é construir histórias de uso de sistemas para que isso sirva de insumo para as equipes de manutenção de software. Os usuários constroem de forma coletiva a história, informando pontos positivos e negativos.

Nesse trabalho foi proposta a ferramenta Feedback, que é uma versão alterada da ferramenta Tellstory.

O trabalho de Carminatti (2006) faz uso do *group storytelling* para reconstruir histórias de acidentes. Essa autora pretende com isso levantar conhecimento tácito que ajude na elaboração de planos de prevenção de acidentes nas organizações. Esse trabalho propõe algumas melhorias com relação à estruturação da história a ser construída, implementando uma versão modificada do Tellstory.

Laporti (2007) sugere a utilização da técnica para fazer levantamento de requisitos de um sistema. A idéia é utilizar o conhecimento coletivo para progressivamente se construir os requisitos do sistema a partir da narrativa. Nesse trabalho, a história construída coletivamente serve de subsídio para construção de cenários que, em seguida, darão origem aos casos de uso.

Reis (2007) usa o *group storytelling* para reconstruir a história das ações tomadas pelas equipes de emergência durante um incidente. Ele pretende dessa maneira identificar ações que obtiveram sucesso e com isso poder categorizar ações resilientes para que as mesmas sejam incorporadas em planos de emergência.

Com exceção do trabalho de Perret, que discute o *group storytelling* de maneira mais geral, os outros trabalhos constituem estudos voltados para aplicação da técnica em situações específicas. Esses trabalhos foram apresentados aqui sem muitos detalhes, mas a seguir observaremos alguns aspectos que darão uma noção melhor de cada um deles.

3.2 Fatores de comparação

Para fazer um estudo mais sistemático das soluções de *group storytelling* consideradas, alguns fatores de comparação foram utilizados. A seguir apresentaremos cada

um desses fatores e nossas respectivas observações. Esse estudo do domínio foi importante para percepção dos problemas considerados nesta dissertação.

3.2.1 Etapas da dinâmica

Ao se observar cada um dos trabalhos sobre o ponto de vista das etapas da dinâmica, é possível fazermos uma separação e uma categorização destas etapas. A dinâmica de *group storytelling* em si muitas vezes é utilizada após uma fase de preparação ou seguida de uma fase onde os resultados da dinâmica servem de insumo para o propósito final do trabalho. Denominaremos as etapas que acontecem antes da dinâmica de pré-dinâmica e aquelas que acontecem após a dinâmica de pós-dinâmica. Com essa separação, é possível evidenciar que muitas vezes, o que é produto de uma fase serve como insumo para uma fase posterior.

Dentro da caracterização de etapas proposta por este trabalho, pode-se notar que o trabalho de Perret (2004) é o único que não cita fases do tipo pré-dinâmica ou pós-dinâmica. É natural que essas etapas apareçam em trabalhos posteriores, que aplicaram a técnica proposta por ele para algum propósito específico.

No trabalho de Oliveira (2006), aparecem apenas duas fases: captura e recuperação. Num primeiro instante poderíamos considerar que a fase de captura seria a dinâmica em si, mas se observarmos os detalhes desta fase podemos notar que existem sub-etapas iniciais que na verdade podem ser classificadas como pré-dinâmica: seleção da atividade, definição da forma de uso do processo e seleção dos participantes. Apenas após essas primeiras tarefas é que começa de fato a dinâmica com participação dos usuários. Já a fase de recuperação acontece após a dinâmica e por este motivo será classificada como uma etapa pós-dinâmica. Nesse trabalho também são evidenciados os produtos produzidos e seus estados possíveis. Assim, ao discutir histórias em construção, histórias em fase de finalização e histórias finalizadas, a autora considera que o produto história passa por cada um destes estados.

A proposta de Carminatti (2006) deixa bem clara a separação entre pré-dinâmica, dinâmica e pós-dinâmica. A fase de preparação é uma pré-dinâmica, a fase de construção é a dinâmica e a fase de publicação é a pós-dinâmica. Na fase de preparação incluem-se todas as tarefas de inicialização da dinâmica e na fase de publicação concentra-se o propósito final do trabalho qual seja o de encontrar nas histórias informações que ajudem na elaboração de planos de prevenção de acidentes nas organizações.

O trabalho de Laporti (2007) também descreve as etapas a serem seguidas detalhando quais os insumos e produtos usados durante todo o processo. Não é considerada uma fase de pré-dinâmica, talvez por não julgar importante essa separação, mas inclui duas fases de pós-dinâmica: uma fase de refinamento das histórias obtidas na primeira fase, com a geração de cenários, e uma fase de conversão destes cenários em casos de uso. Note-se que os produtos e insumos estão bem claros nesse caso. A primeira fase gera as histórias, a segunda fase gera cenários a partir das histórias obtidas na primeira fase e a terceira fase gera casos de uso a partir dos cenários.

A proposta de Reis (2007) é a que apresenta o maior número de fases, sendo uma de pré-dinâmica, uma de dinâmica e quatro de pós-dinâmica. Nesta proposta, nota-se uma relação mais forte entre as fases do processo e o propósito final do trabalho. Na fase de pré-dinâmica existe uma tarefa diretamente ligada ao propósito do processo que é uma fase de caracterização do sistema e dos seus possíveis estados de trabalho. Essa fase é seguida da dinâmica em si, que por sua vez, é seguida de outras fases diretamente ligadas ao objetivo do trabalho: o de identificação de ações resilientes.

A Figura 2 apresenta um mapa conceitual que resume as etapas presentes em cada um dos trabalhos estudados e a classificação dessas etapas em pré-dinâmica, dinâmica ou pós-dinâmica.

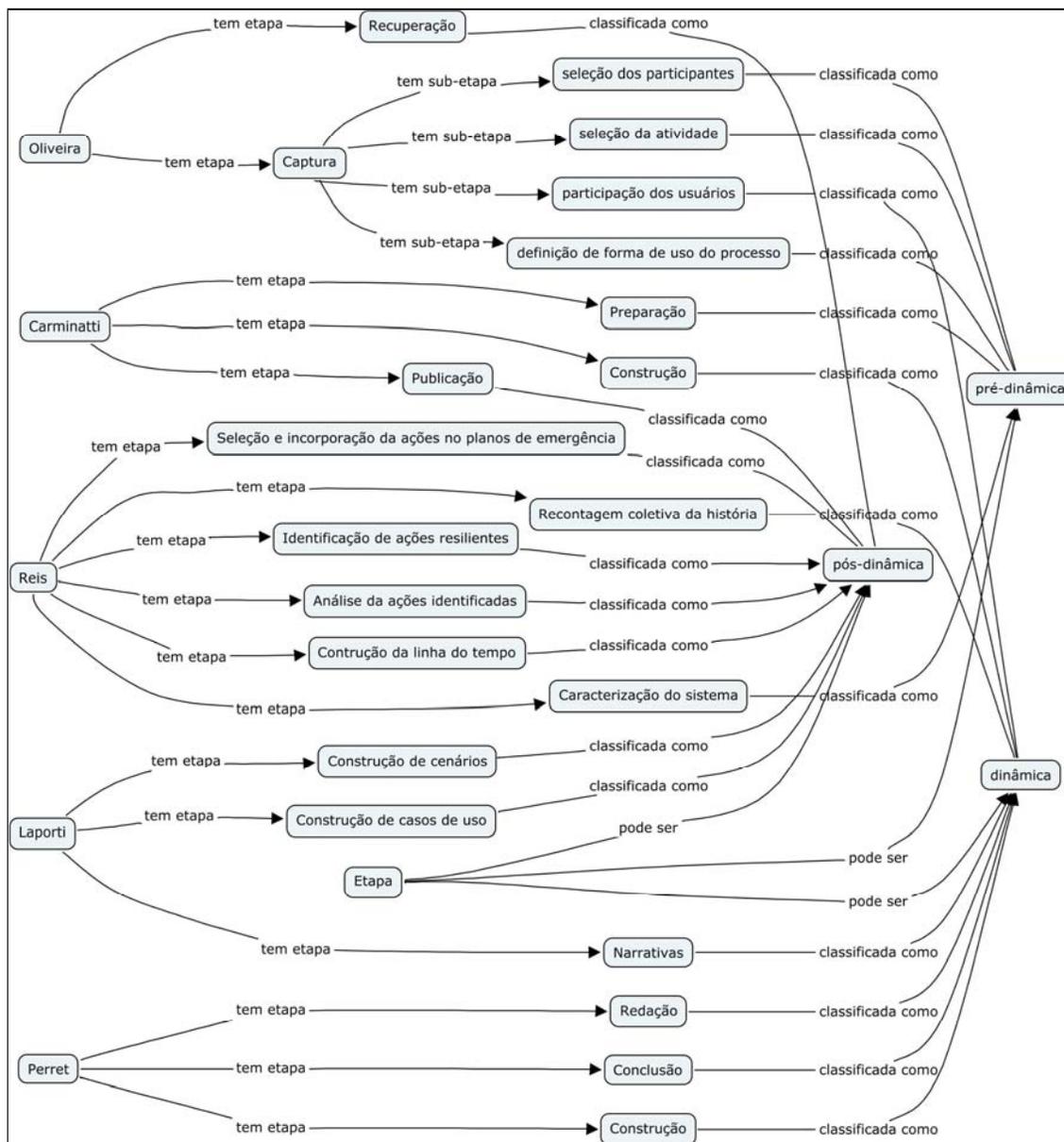


Figura 2 - Análise das etapas da dinâmica de *group storytelling* dos trabalhos estudados

3.2.2 Papéis envolvidos

Ao observarmos os papéis sugeridos por cada um dos trabalhos podemos notar que existem várias semelhanças entre estes. Isto fica claro quando fazemos uma comparação entre as responsabilidades de cada papel e também quando fazemos uma comparação das características e perfis de cada um. A semelhança mais nítida é aquela existente entre os

papéis que são responsáveis por contar a história. O trabalho de Perret (2004) dá a esse papel o nome de "usuário" enquanto que o trabalho de Carminatti (2006) sugere a denominação "contador de histórias". Existe também um papel para quem tem perfil de líder, sendo responsável por conduzir a dinâmica. Nos trabalhos de Perret (2004) e de Laporti (2007) esse papel é denominado de "moderador" enquanto que no trabalho de Carminatti (2006) usa-se a palavra "facilitador". O papel que talvez seja mais importante é aquele exercido pelo indivíduo que irá extrair o conhecimento tácito que foi explicitado nas histórias. Cada um dos trabalhos usou uma denominação distinta para esse papel. É importante lembrar que uma mesma pessoa pode exercer mais de um papel ao mesmo tempo, assim como um papel pode ser exercido por mais de uma pessoa.

A Tabela 2 apresenta os papéis presentes em cada trabalho com as características marcantes de cada um, colocando ainda na mesma linha os papéis equivalentes.

Tabela 2 - Papéis presentes em cada um dos trabalhos de *group storytelling*

Perret	Oliveira	Carminatti	Laporti	Reis	Descrição
usuário	usuários dos sistemas	contador de histórias	stakeholders (usuários, clientes e analistas)	contador de histórias	Papel responsável por contribuir para a construção da história através de depoimentos e interação com os outros participantes.
moderador	moderador	facilitador / coordenador	moderador / comentador	facilitador / coordenador / especialista do domínio	Papel responsável por conduzir a dinâmica de construção da história. Deve estimular os participantes a contribuírem com informações que sejam relevantes para o resultado final.
redator	redator	facilitador	editor	facilitador	Papel responsável por garantir que a história finalizada esteja coerente e bem formatada. Se a história não foi finalizada pelos participantes, esse indivíduo deverá juntar os elementos para redigir a versão final.
comentador	equipes de manutenção	examinador	convertedor	analista de resiliência	Papel responsável por extrair da história todo o conhecimento tácito relevante que foi explicitado.

3.2.3 Estruturação da história

Uma característica importante das histórias é o quanto elas são detalhadas, sobre o ponto de vista estrutural. Isso influencia tanto em sua construção quanto em seu uso para consulta de informações. O detalhamento estrutural refere-se aos pequenos elementos que, juntos, formam a história. Dentre os trabalhos analisados pode-se notar uma variação em

relação a essa questão, com alguns sendo bem específicos em relação aos detalhes estruturais e outros sendo menos objetivos.

Podemos identificar os elementos que formam a história através do estudo da teoria da literatura, com foco nos elementos da narrativa. A maioria dos trabalhos nessa área destacam os seguintes elementos: narrador, personagem, enredo, cenário e tempo. A partir do levantamento dos requisitos de um sistema que vá apoiar a técnica de *group storytelling*, podemos perceber que apenas esses elementos não são suficientes para atender às necessidades dos usuários desse sistema. Por esse motivo, pudemos perceber que os trabalhos aqui estudados propõem diferentes maneiras de estruturar as histórias.

Perret (2004) divide a história em eventos, identificando causas e conseqüências que definem a ordem dos acontecimentos. Além disso, os eventos possuem informações sobre os personagens e as emoções, assim como sobre o ambiente e o período. A Figura 3 apresenta essa estruturação além de algumas outras informações.

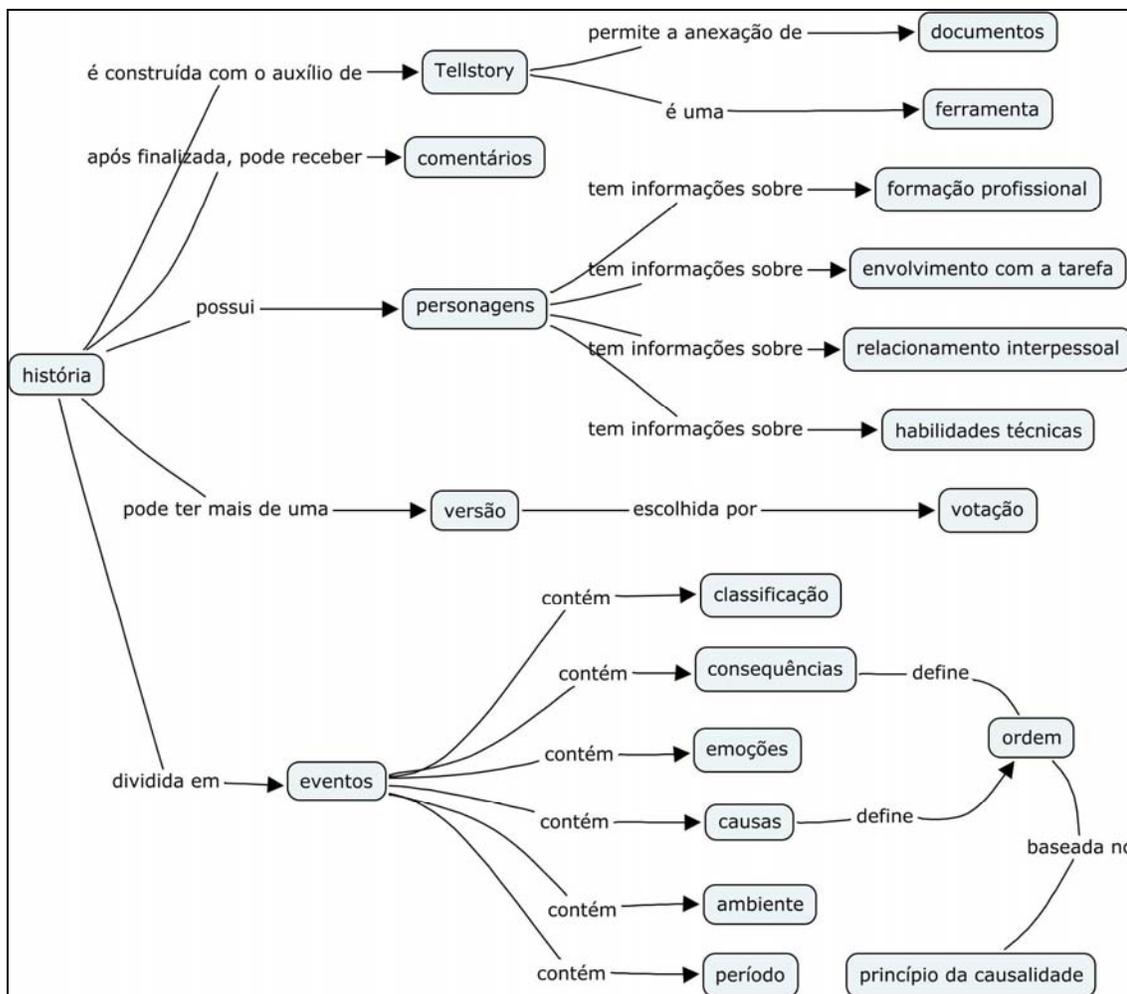


Figura 3 - Estrutura de uma história segundo Perret (2004)

Oliveira (2006) prefere observar a história como uma cadeia de passos. Esses passos contêm informações que podem ser classificadas como problema, benefício, oportunidade, dificuldade, crítica, sugestão ou dúvida, e também ajudam na identificação dos personagens. É a partir das informações contidas nos passos que será identificado o conhecimento tácito. A Figura 4 apresenta essa estrutura.

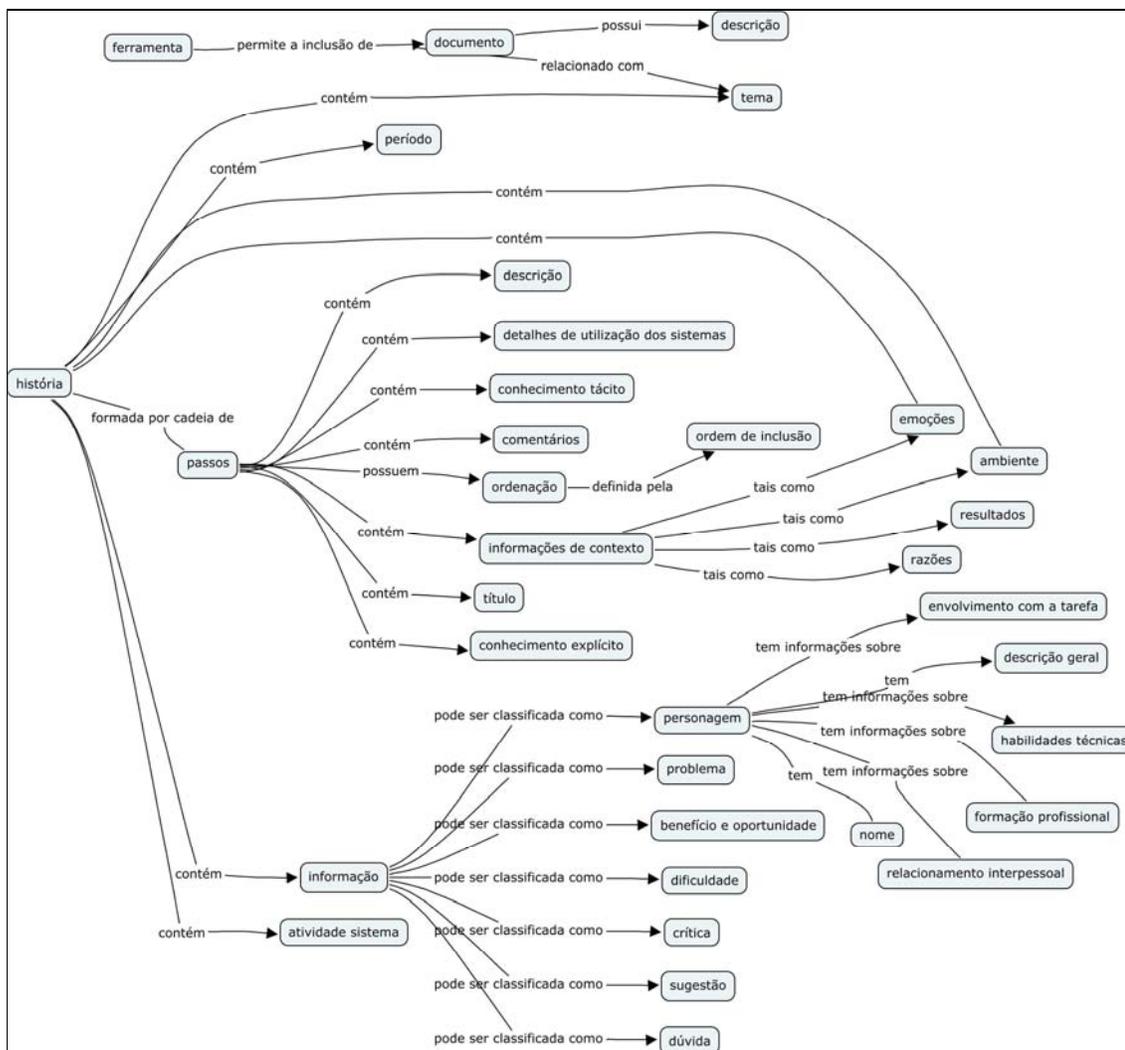


Figura 4 - Estrutura de uma história segundo Oliveira (2006)

Dentre os trabalhos estudados, Carminatti (2006) é o que traz o melhor detalhamento em relação à estrutura da história. Nesse trabalho a história é dividida em fragmentos e esses fragmentos possuem associações. Tanto os fragmentos como as associações podem ser categorizadas. A semântica associada a essas categorias ajuda bastante na organização das informações. Os fragmentos podem ser classificados como fatos, como descritores ou como de relato. Aqueles classificados como de relato podem ainda ser categorizados como: evento, dado, hipótese, opinião ou indefinida. Já os do tipo descritor podem assumir categoria: local, atividade, artefato, personagem ou indefinida. As associações entre fragmentos também têm

categorização e podem ser: causa/consequência, temporal, negação/contradição, confirmação/conformidade, complementação, gap e indefinido. Além destas categorias o trabalho também aceita que novas categorias sejam criadas. A Figura 5 apresenta essa estruturação.

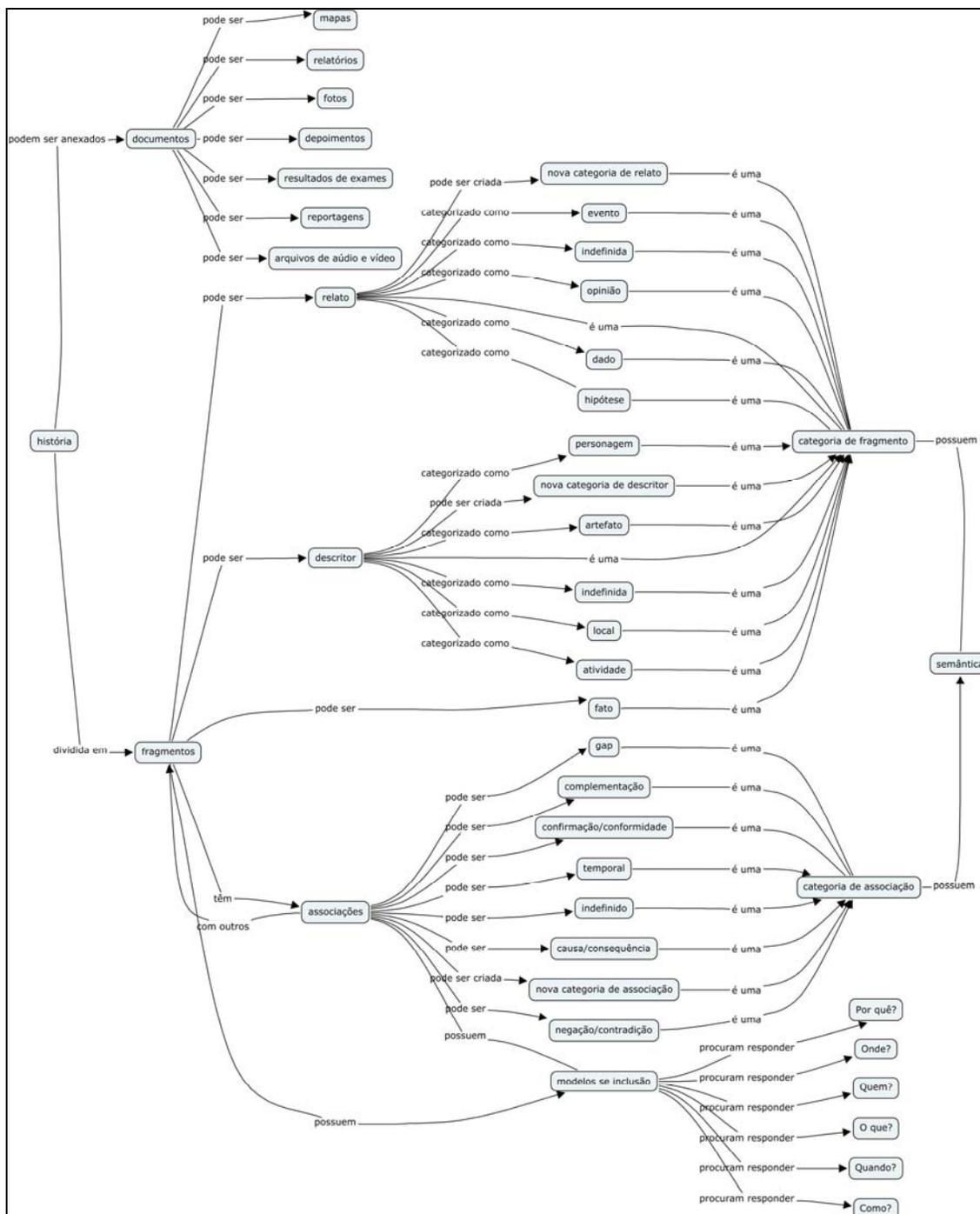


Figura 5 - Estrutura de uma história segundo Carminatti (2006)

Laporti (2007) usa uma abordagem menos estruturada, ou com estruturação evolutiva. Os usuários podem contar a história na forma de texto livre sem imposição de regras ou de formatos pré-definidos. É função do moderador fazer com que as contribuições contenham informações que sejam relevantes. A solução proposta neste trabalho faz uso de um processo dividido em 3 passos, onde no primeiro passo os contadores contam a história na forma de texto livre e nos passos seguintes essas informações ganham a estruturação proposta no trabalho.

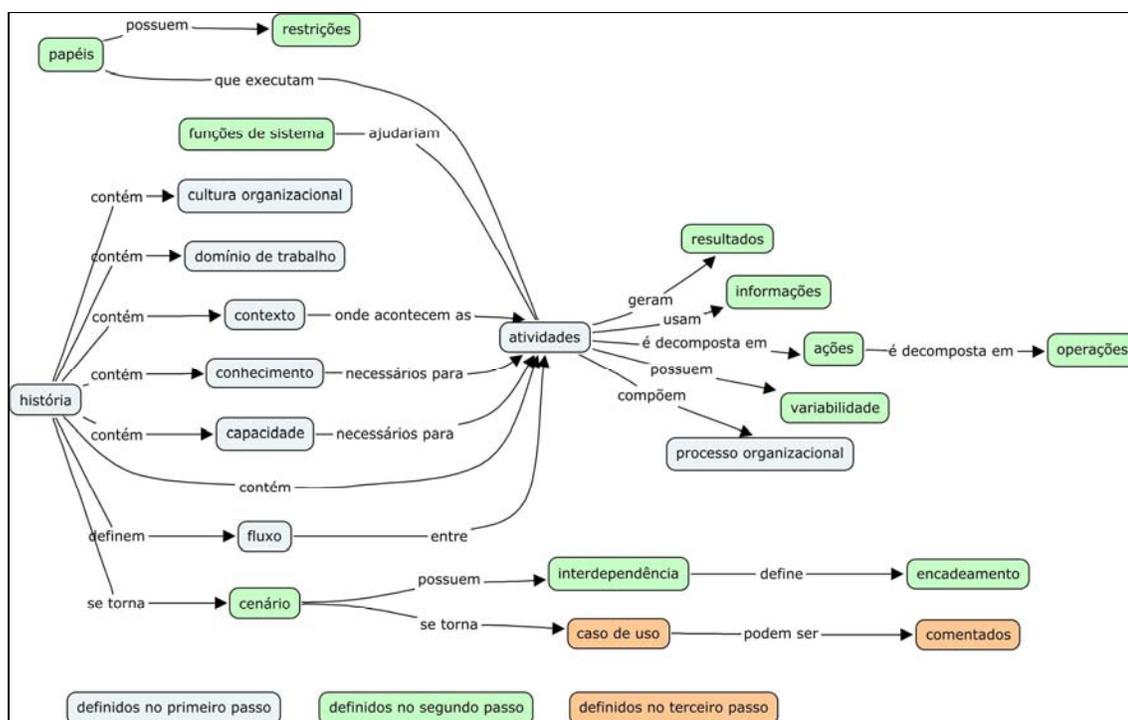


Figura 6 - Estrutura de uma história segundo Laporti (2007)

Reis (2007) usa a mesma estruturação proposta por Carminatti (2006), mas agrega alguns elementos adicionais. Nesse trabalho, a ordem cronológica dos acontecimentos é tratada de forma especial e por isso ele faz uso de linhas do tempo.

Quando se fala de detalhamento estrutural da história, pode-se fazer uma ligação com o uso de modelos de entrada de dados. Quanto mais bem estruturada uma informação, melhor ela se adapta a modelos. Carminatti (2006) cita o uso de modelos de inclusão, tanto para fragmentos das histórias como para associações entre estes fragmentos.

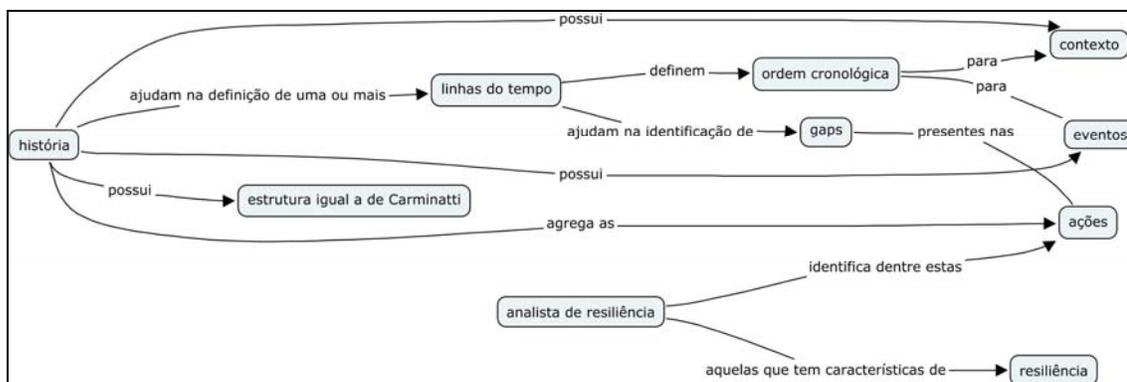


Figura 7 - Estrutura de uma história segundo Reis (2007)

As histórias podem ser classificadas quanto ao seu nível de estruturação. Um texto livre, por exemplo, estaria no nível mais baixo de estruturação. Um texto fragmentado, com associações entre essas partes, já poderia ser considerado como tendo um nível mais alto de estruturação. A questão seria: o excesso de estruturação da história causa impacto na construção e no enriquecimento da mesma? A observação dos trabalhos e das técnicas propostas pelos autores nos faz concluir que sim. Pessoas são diferentes, têm nível de escolaridade diferente, têm formação diferente, e tudo isso influencia na maneira como elas vão lidar com a dinâmica. Uma pessoa com formação na área de exatas lida muito melhor com informações estruturadas do que uma pessoa que seguiu a área de humanas, por exemplo. Para lidar com esse problema alguns trabalhos sugerem que a história deve sofrer uma evolução, ou uma passagem de um nível menos estruturado para um nível mais estruturado, no decorrer da dinâmica. Essa transformação da história pode ser feita por todo o grupo que participa da dinâmica ou por um indivíduo que fica responsável pela fragmentação e

classificação das contribuições. Nesse cenário, as pessoas trabalhariam com textos livres e isso facilitaria a entrada de dados, sem que o processo da dinâmica impusesse nenhum empecilho para as contribuições.

Outra abordagem seria permitir que o próprio usuário pudesse optar pela maneira como quer contribuir. Dessa maneira, uma pessoa que lida melhor com informações estruturadas faria a sua contribuição, possivelmente até com o uso de modelos de entrada de dados, enquanto que outra poderia contribuir com textos livres que posteriormente seriam estruturados. Essa diferenciação pode impactar na questão do trabalho cooperativo já que a contribuição de forma estruturada pode não ser atraente e observável pelos que preferem a não estruturação.

A importância da estruturação surge da necessidade de se trabalhar com essas informações após a finalização da história. O propósito da dinâmica de *group storytelling* é a reconstrução de acontecimentos através do trabalho cooperativo visando à obtenção de conhecimento tácito.

3.2.4 Insumos e produtos gerados

Em todos os trabalhos o insumo principal é a contribuição feita por cada contador da história no formato de texto livre. Perret (2004) reconhece o valor da utilização de áudio e vídeo, mas não faz uso desses artefatos alegando que o esforço pode ser concentrado apenas na narrativa em formato textual. Todos os trabalhos estudados permitem a associação de documentos que possam auxiliar na descrição da história a ser construída. Carminatti (2006) permite a associação de qualquer tipo de arquivo, incluindo fotos e vídeos, mas estes só podem ser associados à história como um todo e não a partes específicas da mesma.

O produto que de forma natural aparece em todos os trabalhos é a história finalizada. A finalização pode ser feita pelos próprios contadores ou por uma pessoa que exerce o papel de construir a versão final a partir das contribuições dos contadores. Alguns trabalhos possuem produtos que estão diretamente ligados ao propósito do estudo. Laporti (2007) trabalha com a história para gerar cenários e, em seguida, casos de uso. Reis (2007) se baseia na história para gerar linhas do tempo e para identificar ações resilientes.

A identificação das etapas que aparecem em cada trabalho também permite identificar os insumos e produtos intermediários de cada etapa. A identificação dos papéis permite que possamos verificar se determinados papéis são responsáveis pela utilização de algum insumo ou pela geração de algum produto.

3.2.5 Quadro comparativo geral

A Tabela 3 e a Tabela 4 apresentam o quadro comparativo geral que faz o cruzamento dos fatores de comparação com os autores de cada um dos trabalhos analisados. Separamos o quadro em duas partes para melhor apresentá-lo nesta seção.

Tabela 3 - Quadro comparativo geral - parte I

		Perret	Oliveira	Carminatti	Laporti	Reis
Etapas	Pré-dinâmica	-	<ul style="list-style-type: none"> • Seleção da atividade • Definição da forma de uso do processo • Seleção dos participantes 	<ul style="list-style-type: none"> • Preparação 	-	<ul style="list-style-type: none"> • Caracterização do sistema e dos seus possíveis estados de trabalho
	Dinâmica	<ul style="list-style-type: none"> • Construção • Redação • Conclusão 	<ul style="list-style-type: none"> • Captura – participação dos usuários 	<ul style="list-style-type: none"> • Construção 	<ul style="list-style-type: none"> • Narrativas 	<ul style="list-style-type: none"> • Recontagem coletiva da história
	Pós-dinâmica	-	<ul style="list-style-type: none"> • Recuperação 	<ul style="list-style-type: none"> • Publicação 	<ul style="list-style-type: none"> • Refinamento – construção de cenários • Conversão – construção de casos de uso 	<ul style="list-style-type: none"> • Construção da linha do tempo • Identificação de ações resilientes • Análise de ações identificadas • Seleção e incorporação de ações nos planos de emergência

Tabela 4 - Quadro comparativo geral - parte II

	Perret	Oliveira	Carminatti	Laporti	Reis
Papéis	<ul style="list-style-type: none"> • Usuário • Moderador • Redator • Comentador 	<ul style="list-style-type: none"> • Usuários dos Sistemas • Moderador • Redator • Equipes de Manutenção 	<ul style="list-style-type: none"> • Contador de Histórias • Facilitador • Coordenador • Examinador 	<ul style="list-style-type: none"> • Stakeholders (Usuários, Clientes e Analistas) • Moderador • Comentador • Editor • Convertedor 	<ul style="list-style-type: none"> • Contador de Histórias • Facilitador • Coordenador • Especialista do domínio • Analista de Resiliência
Estruturação	<ul style="list-style-type: none"> • Dividida em eventos 	<ul style="list-style-type: none"> • Cadeia de passos 	<ul style="list-style-type: none"> • Fragmentos com associações 	<ul style="list-style-type: none"> • Estruturação evolutiva: <ul style="list-style-type: none"> ○ texto livre ○ cenários ○ casos de uso 	<ul style="list-style-type: none"> • Fragmentos com associações • Linha do tempo
Insumos e Produtos	<ul style="list-style-type: none"> • Texto livre • Documentos • História 	<ul style="list-style-type: none"> • Texto livre • Documentos • História 	<ul style="list-style-type: none"> • Texto livre • Documentos • Fotos • Videos • História • Informações que ajudam na elaboração de planos de prevenção de acidentes nas organizações 	<ul style="list-style-type: none"> • Texto livre • Cenários • Casos de uso • História 	<ul style="list-style-type: none"> • Texto livre • Linhas do Tempo • História • Lista de ações resilientes

3.3 Outras soluções

Além dos trabalhos discutidos até aqui, outras soluções que fazem uso de histórias foram analisadas. Algumas delas não utilizam diretamente a técnica de *group storytelling*, mas de alguma maneira usam histórias como tema central de suas abordagens, estimulando e apoiando a prática de se contar histórias em grupo. Vamos apresentar estas soluções sem nos aprofundarmos muito nos detalhes.

Dentre esses outros trabalhos, Silva (2006) também envolve o *group storytelling*. A proposta desta autora é utilizar o *group storytelling* para resolução de conflitos. Ela faz um mapeamento entre os elementos de uma história e as partes de um processo legal. A construção da história vai permitir que sejam identificados argumentos conflitantes ou mesmo de confirmação dentre as partes envolvidas.

Para complementar a análise dos trabalhos do meio acadêmico, algumas soluções comerciais disponíveis também foram estudadas. A implementação disponível em StoryStream (2008) permite que usuários criem histórias e possam compartilhá-las com outros usuários. As histórias recebem notas e aquelas com melhores avaliações recebem destaque na página principal do sistema. É possível classificar as histórias em comédia, drama, romance, dentre outras categorias.

Dandelife (2007) usa uma abordagem um pouco diferente, pois a idéia neste caso é permitir que os usuários criem histórias sobre suas vidas, como uma biografia. O ponto forte desta solução é sua integração com outros sistemas disponíveis na Internet o que permite que bastante conteúdo seja adicionado à história de um usuário sem sobrecarregar o sistema.

Já em OurStory (2007), a solução foi desenvolvida com foco em linhas do tempo. Nesse sistema, o primeiro passo que um usuário deve executar para criar uma história é definir a sua linha do tempo. Em seguida ele convida outros usuários para criarem a história

por sobre a linha do tempo de forma compartilhada. O sistema também permite que fotos e vídeos sejam utilizados na construção da história.

A análise desses trabalhos foi importante para observarmos soluções e abordagens diferentes daquelas adotadas pelos trabalhos estudados com maior profundidade. A utilização de outras mídias e a integração entre diferentes sistemas foi o grande diferencial apresentado aqui.

3.4 Problemas com as constantes re-implementações

A análise das variadas soluções de apoio ao *group storytelling*, assim como de outras aplicações que têm como foco a criação de histórias, nos fez refletir sobre os problemas enfrentados durante o desenvolvimento destes sistemas. As constantes re-implementações, ou mesmo as adaptações que se seguiram uma após a outra, indicaram a necessidade de criação de uma plataforma base que facilitasse esse desenvolvimento.

Uma das características das soluções desenvolvidas era a falta de flexibilidade que elas ofereciam para possíveis alterações que deveriam ser feitas no intuito de atender um requisito específico. Um desenvolvedor que quisesse fazer uma alteração no sistema acabava afetando outras partes não ligadas ao trecho que queria modificar.

Além disso, as soluções tinham pouca ou nenhuma integração com outras fontes de conteúdos, o que impedia a expansão para além da fronteira do próprio sistema. A possibilidade de conteúdos construídos dentro do sistema serem associados com outros conteúdos presentes em locais fora dali nunca foi uma alternativa presente nessas soluções.

É natural que essas soluções deixem a desejar na questão da arquitetura do software, em termos de flexibilidade, pois o foco da implementação era resolver os requisitos da situação em questão, e não atender a possíveis variações destes requisitos. No entanto, quando um sistema é desenvolvido de forma modular, com componentes bem definidos e

categorizados, e com a adoção de padrões, a compreensão e o entendimento daquela solução são facilitados. Isso permite que especializações daquela solução sejam construídas mais facilmente além de possibilitar uma evolução mais suave do software. Um arcabouço pareceu ser a solução ideal para estes problemas.

4 Arcabouços

Neste capítulo apresentaremos arcabouços de forma geral, falando sobre algumas definições encontradas na literatura, apresentando metodologias de desenvolvimento e fazendo uma comparação entre elas, observando diferentes maneiras de classificar arcabouços e finalizando com um olhar sobre o ciclo de vida dos arcabouços.

4.1 Arcabouço como solução

Um *framework*, ou um arcabouço, palavra usada nessa dissertação e encontrada em textos mais recentes em português, pode ser descrito genericamente como uma estrutura sobre a qual algo será construído. O arcabouço que será proposto neste trabalho visa ser a base sobre a qual soluções de apoio ao *group storytelling* serão construídas.

Johnson e Foote (1988) definem um arcabouço como um conjunto de classes abstratas aplicadas como solução para uma família de problemas relacionados, e que suporta reuso num nível de granulosidade maior do que o de classes. Gamma *et al.* (1995) seguem a mesma linha dizendo que um arcabouço é um conjunto de classes que cooperam entre si formando um projeto reutilizável para um segmento de software específico.

Buschmann *et al.* (1996) consideram um arcabouço, num contexto de arquitetura, como um sistema de software parcialmente completo cujo propósito é ser instanciado. Ele define a arquitetura para uma família de sistemas e fornece os blocos básicos de construção para criá-los. Ele também especifica os locais onde adaptações para funcionalidades específicas devem ser feitas. Num ambiente OO, um arcabouço consiste em classes abstratas e concretas.

Fayad e Schmidt (1997) afirmam que um arcabouço é uma aplicação “semi-completa”, reutilizável, que pode ser especializada para se produzir aplicações customizadas.

Eles citam 4 principais vantagens no uso de arcabouços por parte dos desenvolvedores: modularidade, reusabilidade, extensibilidade e inversão de controle.

Para Larman (2002), um arcabouço é um conjunto extensível de objetos para funções relacionadas. Ele fornece uma implementação para as funções centrais e invariáveis, e inclui um mecanismo que permite que um desenvolvedor encaixe as funções variáveis ou estenda as funções existentes.

Existem diversas outras definições na literatura com algumas seguindo um raciocínio mais técnico e detalhista e outras preferindo conceitos mais abrangentes, como os de Buschmann *et al.* (1996) e de Larman (2002), que são os que adotaremos nessa dissertação. O arcabouço que será proposto nesse trabalho é um software parcialmente completo. As funcionalidades centrais estarão implementadas e um passo a passo de complementação será apresentado para instanciação do arcabouço.

4.2 Metodologias de desenvolvimento de arcabouços

O estudo de algumas metodologias de desenvolvimento de arcabouços presentes na literatura nos ajudou a identificar atividades essenciais que deveriam ser executadas para que pudéssemos chegar a uma proposta de arcabouço. Essas metodologias têm muitos pontos em comum e, em geral, seguem um roteiro de estudo do domínio, seguido da construção do arcabouço e de uma fase de testes.

Markievicz e Lucena (2007) fazem um paralelo entre as atividades do processo de desenvolvimento de aplicações OO e as atividades do processo de desenvolvimento de um arcabouço. Enquanto para se desenvolver uma aplicação o levantamento de requisitos estuda um único problema, para se desenvolver um arcabouço deve-se estudar um domínio como um todo. Além disso, o resultado final no primeiro caso é uma aplicação executável enquanto que no segundo caso são várias instâncias do arcabouço. A Figura 8 apresenta essa comparação.

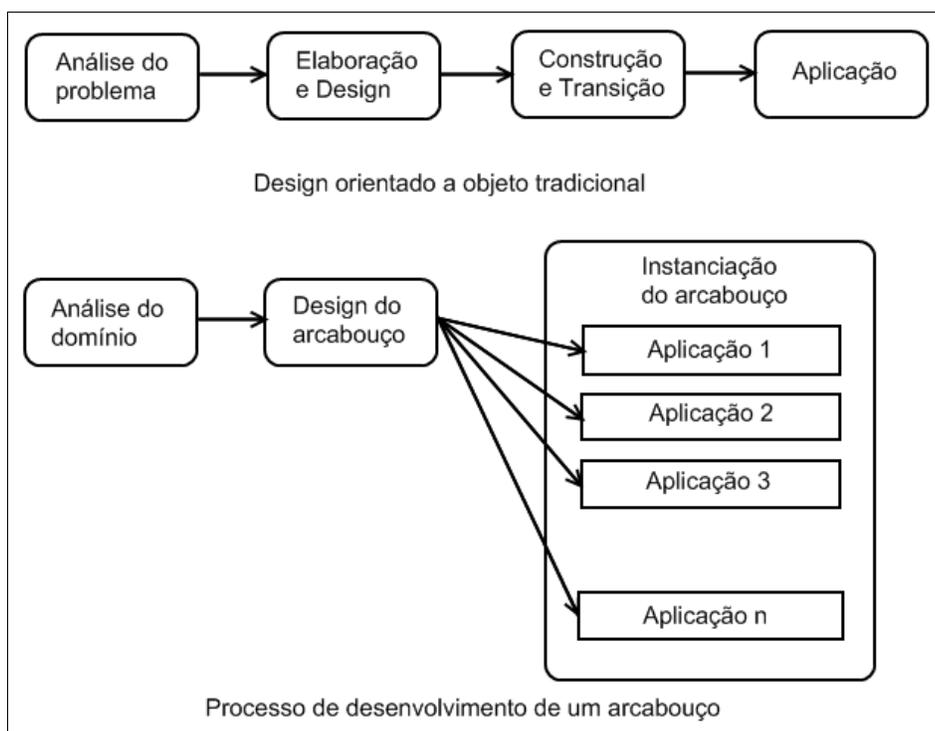


Figura 8 - Comparação entre design OO tradicional e processo de desenvolvimento de arcabouços - extraído de Markiewicz e Lucena (2007)

Silva (2000) apresenta um estudo comparativo bem completo de 3 metodologias de desenvolvimento de arcabouços. Ele analisa os pontos em comum e as características específicas das metodologias: Projeto Dirigido por Exemplo [Example-driven Design] (JOHNSON, 1993), Projeto dirigido por Hot Spot [Hot Spot driven design] (PREE, 1995) e a metodologia de projeto da empresa Taligent (TALIGENT, 1994).

O processo de desenvolvimento de um arcabouço, segundo o Projeto Dirigido por Exemplo, atravessa as etapas de análise, projeto e teste, de acordo com a Figura 9. O diferencial dessa metodologia é a ênfase na análise do maior número possível de aplicações já desenvolvidas. É desta análise que vão surgir os pontos em comum e conseqüentemente as características do domínio, assim como as diferenças que irão originar os pontos de flexibilidade do arcabouço.

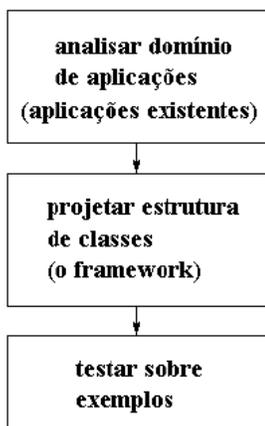


Figura 9 - Etapas do Projeto Dirigido por Exemplo para o desenvolvimento de um arcabouço - extraída de (SILVA, 2000)

O que diferencia uma aplicação orientada a objetos de um arcabouço é que a primeira é completamente definida enquanto que o segundo possui partes propositalmente indefinidas que garantirão a flexibilidade para ele se tornar uma aplicação específica. Os *Hot Spots* são essas partes mantidas flexíveis. A principal característica dessa metodologia é a identificação desses pontos de flexibilidade na estrutura de classes de um domínio para, a partir daí, se construir o arcabouço. A Figura 10 apresenta as etapas para o desenvolvimento de um arcabouço segundo a metodologia de Pree (1995).

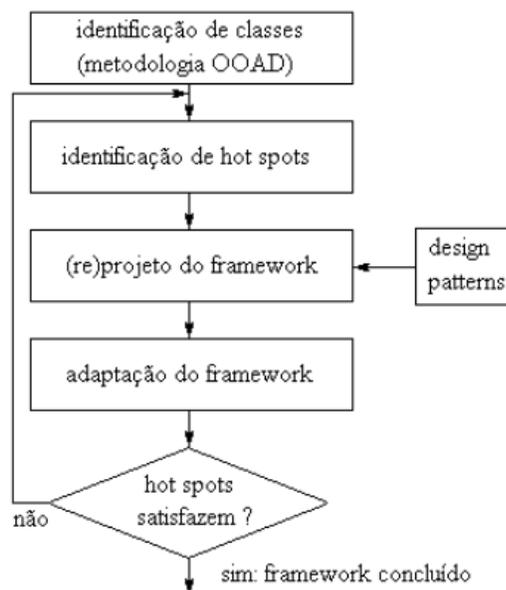


Figura 10 - As etapas do Projeto Dirigido Por Hot Spot para o desenvolvimento de um arcabouço - extraída de (SILVA, 2000)

A metodologia de projeto da empresa Taligent é norteada por dois princípios: produzir um conjunto de arcabouços estruturalmente menores e mais simples, que, usados conjuntamente, darão origem às aplicações, e tornar o uso do arcabouço o mais simples possível através da minimização da quantidade de código que o usuário deve produzir. Essa metodologia sugere a seqüência de 4 passos:

1. Identificar e caracterizar o domínio do problema;
2. Definir a arquitetura e o projeto;
3. Implementar o arcabouço;
4. Desdobrar o arcabouço.

Outra metodologia estudada foi a apresentada por Yacoub e Ammar (2000), onde os arcabouços são definidos a partir de padrões de projeto (*Pattern-Oriented Frameworks*). O processo de desenvolvimento apresentado por eles segue uma seqüência de 6 passos:

1. Análise do sistema;
2. Identificação de subsistemas;
3. Análise dos subsistemas;
4. Instanciação e expansão de padrões;
5. Redução;
6. Agrupamento.

Através do estudo destas diferentes metodologias foi possível identificar características em comum entre elas. Todas sugerem o estudo sobre o domínio em questão através da análise de aplicações já elaboradas. Além disso, o uso de padrões de projeto foi unanimidade no intuito de melhorar o desenvolvimento. Também foi consenso que uma última etapa de testes através da instanciação do arcabouço é necessária para se considerar fechada uma iteração de desenvolvimento.

As metodologias não foram apresentadas com todos os seus detalhes, pois este não é o intuito deste trabalho. Os leitores que desejarem se aprofundar devem seguir as referências apresentadas no decorrer do texto.

4.3 Classificação de arcabouços

Para compreendermos melhor o arcabouço que será proposto nessa dissertação iremos situá-lo de acordo com dois tipos de classificação presentes na literatura. O primeiro deles, apresentado por Johnson e Foote (1988), é o que distingue arcabouços em caixa-branca e caixa-preta. Com essa classificação podemos saber como o arcabouço é utilizado. O segundo, apresentado por Fayad e Schmidt (1997), nos ajuda a identificar onde o arcabouço é aplicado. Nessa classificação um arcabouço pode se enquadrar nas categorias: infra-estrutura de sistema, *middleware* de integração ou aplicação.

4.3.1 Quanto à técnica de utilização

O conceito de caixa-branca e caixa-preta é o mesmo que existe na área de teste de software. A analogia com uma caixa branca segue a idéia de que uma caixa branca seria transparente permitindo com isso que seu interior fosse visto. Uma caixa preta, por outro lado, não é transparente e seu interior não pode ser visto.

Arcabouços caixa-branca são aqueles que exigem que o desenvolvedor que irá utilizá-los saiba como é o seu funcionamento interno. Esses desenvolvedores terão acesso ao código fonte ou então usarão herança para criar variações das classes base. A necessidade de conhecimento da implementação do arcabouço implica que ele seja muito bem documentado, inclusive com exemplos de instanciação. Além disso, a curva de aprendizado para que seja bem utilizado é longa.

O outro tipo de arcabouço segundo essa classificação é o caixa-preta. Nesse caso, a instanciação do arcabouço e a configuração do comportamento da aplicação são feitas pelo encaixe de diversos componentes. Cada um dos componentes trabalha com um tipo específico de protocolo e isso faz com que os desenvolvedores que forem utilizar o arcabouço tenham que saber apenas como é a interface externa desses componentes.

Alguns autores como Earles (2007) e Markievicz e Lucena (2007) usam o conceito de caixa-cinza para classificar arcabouços híbridos, os quais têm características tanto de caixa-branca como de caixa-preta. O primeiro, inclusive, considera que arcabouços caixa-branca evoluem para caixa-preta conforme ganham maturidade.

4.3.2 Quanto ao escopo de utilização

Fayad e Schmidt (1997) citam três categorias de escopo de utilização de arcabouços que nos ajudam a compreender melhor onde esses artefatos podem ser aplicados.

Arcabouços de infra-estrutura de sistema servem de apoio para o desenvolvimento de sistemas operacionais e de comunicação, interfaces de usuário e ferramentas de processamento de linguagem. Na maioria das vezes esses arcabouços são utilizados internamente em organizações que desenvolvem software e não são vendidos para terceiros.

Arcabouços do tipo *middleware* de integração são usados para integrar aplicações e componentes distribuídos. Ajudam os desenvolvedores de software no trabalho relacionado à modularização e reuso em sistemas que atuam em ambiente distribuído. Enquadram-se nessa categoria arcabouços ORB, *middleware* orientado a mensagens e bancos de dados transacionais.

Arcabouços de aplicação são usados para apoio a domínios específicos como telecomunicações, aviação ou área financeira. Se comparados aos outros tipos de arcabouço, o desenvolvimento de arcabouços dessa categoria é caro, porém, o ganho na criação de novos sistemas é imediato.

4.4 Ciclo de vida de um arcabouço

Arcabouços têm um ciclo de vida que pode ser separado em duas fases, que não necessariamente estão isoladas uma da outra: construção e reuso. Essas fases podem ser sobrepostas, já que a evolução de um arcabouço se dá justamente pelas avaliações que são feitas conforme ele está sendo usado.

Inicialmente, o desenvolvimento do arcabouço é disparado por um conceito freqüente recorrente, que constitui a motivação para a construção do arcabouço, o propósito para o qual ele serve, e em qual domínio ele se aplica (YACOUB; AMMAR, 2000). A partir desse instante se inicia a fase de construção. Essa fase contém atividades comuns a qualquer desenvolvimento de software: *design*, codificação, teste, documentação, etc.

A segunda fase, o reuso do arcabouço, já acontece na etapa de testes, mas só consideramos realmente como reuso quando o arcabouço está sendo instanciado para

construção de aplicações específicas. É essa fase que vai trazer informações para que o arcabouço possa evoluir e ganhar maturidade.

No estudo feito nesse trabalho cobrimos o ciclo de vida do arcabouço parcialmente já que executamos uma primeira iteração da fase de construção e fizemos algumas validações de reuso. Conforme outras pessoas utilizem o arcabouço surgirão novas demandas e outras iterações da fase de construção se farão necessárias.

5 Sofia - Conceção

O objetivo deste capítulo é apresentar o arcabouço proposto sem entrar nos detalhes de implementação. Iniciaremos falando sobre a metodologia que utilizamos para chegar à proposta e em seguida apresentaremos as características da solução. Nessa apresentação, vamos inicialmente observar o arcabouço sob os mesmos aspectos utilizados no capítulo 3 e em seguida apresentaremos as funcionalidades organizando-as segundo o modelo 3C de *groupware*. Ainda neste capítulo apresentaremos os pontos de flexibilidade e variabilidade oferecidos pelo arcabouço. Finalizaremos classificando o arcabouço segundo os critérios apresentados no capítulo 4.

5.1 Um arcabouço como proposta de solução

A constante re-implementação de soluções para apoio ao *group storytelling* nos fez refletir sobre se seria possível desenvolver algum artefato que pudesse agregar os pontos em comum ao mesmo tempo em que permitisse a extensão com funcionalidades específicas. O propósito final seria evitar o re-trabalho.

Muitos desenvolvedores de software buscam a reutilização de código com o intuito de poupar tempo e esforço. Essa reutilização vai desde o nível do método, passando por classes e padrões de projeto, e chega até o nível de arcabouços. Esse último sendo bem mais complexo e sofisticado, pois boa parte de decisões de projeto fica embutida nele.

O estudo e a análise detalhados dos trabalhos anteriores que fazem uso da técnica de *group storytelling* permitiram que identificássemos os pontos em comum e que, a partir daí, surgisse a idéia de se propor um arcabouço como solução para o desenvolvimento de software de apoio *group storytelling*.

No entanto, apenas o estudo dos trabalhos já desenvolvidos não seria suficiente para se chegar a um arcabouço com as características que gostaríamos. Outros aspectos tiveram que ser observados e partimos para o estudo de sistemas de *groupware*. Esses sistemas se caracterizam por serem softwares projetados para serem usados de forma colaborativa por várias pessoas e, “refletem uma mudança na ênfase de se usar o computador para resolver problemas, em usar o computador para facilitar a interação humana” (ELLIS, 1991). Sistemas de apoio ao *group storytelling* se encaixam nessa categoria.

Além dessas abordagens também foi necessário um estudo das metodologias de desenvolvimento de arcabouços para que pudéssemos definir a nossa metodologia e pudéssemos chegar a uma proposta de arcabouço. A nossa proposta está baseada em 3 linhas de estudo:

- Análise de trabalhos já elaborados com *group storytelling*;
- Estudo sobre sistemas de *groupware*;
- Melhores práticas de construção de arcabouços.

As duas primeiras linhas de estudo foram discutidas no capítulo 3 quando fizemos uma análise dos trabalhos que já tinham feito propostas de soluções para serem usadas com o *group storytelling* e a terceira linha de estudo foi discutida no capítulo 4 onde procuramos fazer um levantamento sobre arcabouços em geral.

O sucesso de um arcabouço é proporcional à quantidade de vezes que ele é utilizado, então, para facilitar o entendimento por parte daqueles que forem utilizá-lo, procuramos adotar sempre conceitos bem difundidos e estabelecidos dentro da ciência da computação.

5.2 Abordagem deste trabalho

A metodologia de construção do arcabouço adotada por este trabalho seguiu a linha geral presente nas outras metodologias. As etapas de estudo do domínio, construção do

arcabouço e testes foram executadas, sendo que cada uma delas teve características específicas adotadas por alguma motivação especial.

Na etapa de estudo do domínio fizemos a análise de algumas aplicações que já tinham implementado uma solução de TI para apoio ao *group storytelling*. Essa análise foi essencial para identificarmos os pontos em comum e a terminologia a ser adotada no nosso arcabouço. Além disso, também utilizamos uma estratégia *top-down* de análise de requisitos de sistemas de *groupware*. Esse segundo passo dentro do estudo do domínio foi importante para identificarmos requisitos ainda não atendidos pelas aplicações já desenvolvidas e também foi importante para ser um guia na modularização e componentização do arcabouço.

A etapa de construção do arcabouço foi em parte baseada nas implementações estudadas e em parte baseada em conceitos de engenharia de software bem difundidos. As implementações estudadas foram a base para a modelagem de dados e para idéias de interface com usuário. Por outro lado, as macro e micro arquiteturas do arcabouço foram todas implementadas usando conceitos de padrões de projeto, além de tecnologia de componentes.

A etapa de testes foi feita com a instanciação do arcabouço para a criação de duas aplicações. A primeira delas foi feita com o intuito de se medir de forma objetiva o esforço necessário para instanciação e a segunda para se obter um relatório de *feedback* (avaliação subjetiva) sobre esse mesmo esforço. Os detalhes destes testes estão apresentados no capítulo 7.

Se fizermos um paralelo entre nossa abordagem e as metodologias estudadas no capítulo 4, notaremos que adotamos práticas presentes em cada uma delas. Baseamo-nos no maior número de exemplos como sugere a metodologia de Johnson (1993), tentamos identificar pontos de flexibilidade conforme a metodologia de Pree (1995), procuramos gerar um arcabouço que exigisse pouca codificação por parte dos instanciadores, seguindo o que

determina a metodologia da empresa Taligent (1994), e procuramos utilizar padrões de projeto que são o foco da metodologia proposta por Yacoub e Ammar (2000).

5.3 Características herdadas das soluções estudadas

A implementação do arcabouço procurou equilibrar e atender os diversos requisitos que existiam nas soluções estudadas. Vamos procurar fazer um paralelo entre o que foi discutido no capítulo 3 e o que está presente na versão atual do arcabouço.

5.3.1 Etapas da dinâmica

No capítulo 3 procuramos classificar cada uma das etapas presentes nos trabalhos estudados como pré-dinâmica, dinâmica ou pós-dinâmica, para tentar compreender quais delas faziam parte do *group storytelling* diretamente e quais eram fases de preparação ou de análise dos resultados. Com essa classificação foi possível percebermos que ocorriam as mais diversas situações: enquanto um trabalho só tratava da dinâmica, outros davam foco maior para pós-dinâmica, e outros para pré-dinâmica.

O arcabouço procura atender às três categorias de etapa, mas certamente com mais ênfase a categoria chamada de dinâmica, que representa a realização do *group storytelling* realmente. Entretanto, antes que os usuários comecem a cooperar para construir a história existe uma fase de convocação destes usuários que é feita também com o auxílio do arcabouço e, portanto, podemos considerar que o arcabouço cobre pelo menos essa atividade que seria classificada como pré-dinâmica.

A fase da dinâmica em si é a melhor atendida pelo arcabouço, pois os requisitos de cooperação exigidos aqui são comuns à maioria dos trabalhos e foi possível implementá-los de modo que atendessem a todos. Criação e associação de fragmentos, escrita de comentários,

inclusão de mídias, são alguns exemplos de funcionalidades presentes no arcabouço que atendem a requisitos exigidos nesta etapa.

As atividades que se classificam como pós-dinâmica são aquelas que começam necessariamente após a finalização de uma história, ou seja, quando não se pode mais incluir peças na história. O arcabouço não tem funcionalidades que atendem a requisitos desta categoria de etapa e é natural que ele seja assim já que a análise das histórias é algo específico de quem irá instanciar o arcabouço. No entanto, uma boa interface de leitura das histórias finalizadas foi um requisito que procuramos atender principalmente pela característica de hipertexto das histórias construídas no ambiente, com fragmentos interligados por associações.

5.3.2 Papéis envolvidos

No estudo sobre os papéis que as pessoas exercem quando estão envolvidas com a dinâmica de *group storytelling* pudemos perceber que existia grande semelhança entre as atividades exercidas por essas pessoas, mas cada trabalho optou por nomear esses papéis de maneira própria. A Tabela 2, que está presente no capítulo 3, apresenta um quadro com esses papéis e suas equivalências.

O arcabouço proposto nesta dissertação trabalha com papéis que se originaram deste estudo e ele adota uma nomenclatura única e mais simplificada. No lugar de 4 papéis, como nos trabalhos estudados, o arcabouço enxerga apenas 3, que exercem toda a gama de atividades envolvidas na dinâmica. São eles: contador, facilitador e leitor.

O **contador** é o papel responsável pela construção da história. Ele coopera com outros contadores durante o processo de escrita da história. É essa pessoa que vai escrever os fragmentos, fazer associações entre fragmentos, comentar fragmentos de outros contadores, adicionar outros tipos de conteúdos à história, dentre outras atividades colaborativas.

O **facilitador** é o papel responsável pela coordenação do processo de construção da história. Um usuário que cria uma nova história se torna automaticamente o facilitador desta história. Essa pessoa deve se preocupar com o andamento do processo de construção e também deve estimular os contadores a contribuírem com conteúdos que sejam relevantes.

O papel de **leitor** é exercido por aquelas pessoas que vão extrair das histórias criadas todo o conhecimento que lhes interessa. É importante, principalmente para essas pessoas, que a história contenha bastante conteúdo relevante, então uma interação entre leitores e facilitadores deve ser encorajada.

Além destes papéis que estão diretamente envolvidos com a dinâmica ainda existe o papel de **administrador**. Usuários com este perfil podem fazer alterações que impactam toda uma instância do arcabouço.

5.3.3 Estruturação da história

De maneira geral, a estrutura das histórias segundo os trabalhos estudados é composta por partes que têm algum tipo de associação entre si. Cada um dos trabalhos usou uma denominação diferente para essas partes (evento, passo, fragmento), mas a função destes elementos era a mesma, ou seja, eles se associavam uns com os outros para formar a história.

O trabalho mais completo em termos de estruturação das histórias foi o apresentado por Carminatti (2006) e grande parte do que foi implementado no arcabouço herdou as características desta solução. Histórias são compostas por fragmentos que têm associações entre si. Os fragmentos são classificados em tipos que se encaixam em alguma categoria. Associações entre fragmentos também são classificadas por tipos. Essa estrutura está resumida na Figura 11.

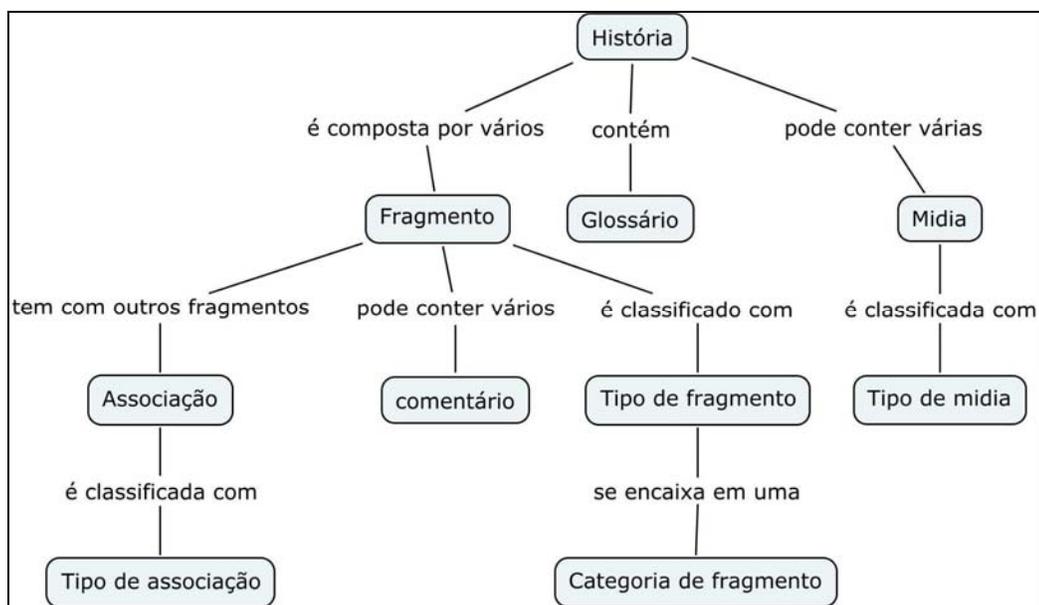


Figura 11 - Estrutura de uma história no arcabouço

Além dos fragmentos e das associações outros elementos complementam a construção de uma história. Comentários podem ser adicionados a fragmentos. Esse é um recurso utilizado pelos contadores para agregar informações a um fragmento. Toda história possui um glossário e os contadores podem incluir novos termos neste glossário.

Um elemento de complementação que foi discutido nos outros trabalhos, mas que não foi implementado em boa parte deles foi a possibilidade de se incluir outras mídias na história, além de apenas texto. O arcabouço trata outras mídias com a mesma importância da mídia texto e permite que os contadores possam agregar elementos como foto, áudio ou vídeo, além de qualquer outro tipo de conteúdo disponível na Internet.

Os fragmentos podem ser classificados com o uso de tipos e categorias, as associações entre os fragmentos podem ser classificadas usando tipos, e as mídias também podem ser classificadas através de tipos. Existem valores padronizados para esses elementos, mas o arcabouço permite que o facilitador da história altere as opções disponíveis para os contadores. Esse tipo de configuração também pode ser feita por um usuário com perfil de

administrador, mas as alterações feitas por ele vão afetar todo o sistema e não apenas uma história.

5.3.4 Insumos e produtos

Conforme feito no capítulo 3, podemos observar os insumos e produtos trabalhados pelo arcabouço associando-os com etapas e com papéis exercidos por pessoas. Na pré-dinâmica apenas o facilitador está envolvido e não existem muitos insumos ou produtos a se considerar. As etapas da dinâmica e da pós-dinâmica são as que mais trabalham com estes artefatos.

Durante a dinâmica, todo tipo de contribuição adicionada na história pelos contadores pode ser considerada como insumo. Fragmentos, comentários, associações, mídias e termos de glossário estão nesse grupo. Quando a história é finalizada pelo facilitador podemos dizer que um produto foi gerado. Uma história finalizada é um produto da etapa denominada dinâmica.

Após o fim desta etapa, quando algum leitor vai utilizar a história para extração de conhecimento, dá-se início à etapa pós-dinâmica. Nesse momento a história finalizada passa a ser insumo e novos produtos serão gerados. Esses novos produtos podem ser quaisquer artefatos que utilizam o conhecimento extraído das histórias e faz parte do processo de instanciação do arcabouço a definição dos mesmos.

5.4 Funcionalidades presentes no arcabouço classificadas segundo o modelo 3C

5.4.1 *Groupware* baseado no modelo 3C

Para classificarmos cada uma das funcionalidades presentes no arcabouço utilizamos o modelo 3C. Esse modelo foi proposto por Ellis *et al.* (1991), mas adotamos o modelo assim como apresentando por Fuks *et al.* (2005), que difere do modelo de Ellis apenas na terminologia. A colaboração é vista como a combinação de 3 elementos: comunicação, coordenação e cooperação. Como estamos falando de interação dentro de um grupo de pessoas, a comunicação significa a troca de mensagens entre essas pessoas, a coordenação significa a gestão destes indivíduos e de suas atividades, e a cooperação, o trabalho feito em conjunto num espaço compartilhado.

A utilização desse modelo, além de nos ajudar nesta classificação, nos permitiu fazer uma análise do domínio de forma mais detalhada e objetiva. O estudo dos trabalhos anteriores de *group storytelling* foi feito com foco nos elementos comunicação, coordenação e cooperação. Essa separação nos ajudou a enxergar o arcabouço de forma modularizada, com os componentes sendo estabelecidos dentro de cada um desses elementos.

Considerando o elemento comunicação, uma taxonomia que nos ajuda a subdividir essa categoria é aquela que relaciona tempo e espaço. A combinação dos valores “mesmo local” e “locais diferentes”, para o eixo espaço, com os valores “mesmo tempo” e “tempos diferentes” do eixo tempo, nos leva aos quadrantes apresentados na Tabela 5.

Tabela 5 - Taxonomia tempo x espaço para o elemento comunicação

	Mesmo tempo	Tempos diferentes
Mesmo local	Interação face-a-face	Interação assíncrona
Locais diferentes	Interação distribuída síncrona	Interação distribuída assíncrona

Softwares de mensagens instantâneas, correio eletrônico, fórum de discussão, blog, tele-conferência, vídeo-conferência, lista de discussão são alguns dos mecanismos de

comunicação disponíveis atualmente para usuários de TI. Esses mecanismos necessariamente se encaixam em um dos quadrantes da tabela apresentada.

A utilização ou não deles depende do propósito para o qual um sistema está sendo construído. No caso do *group storytelling* alguns desses mecanismos não são bem-vistos e a justificativa para isso é que o uso deles faria com que os participantes registrassem o conhecimento fora dos fragmentos da história ao estarem usando essas outras funcionalidades.

O controle desse problema e de outros que surgem em trabalhos colaborativos é feito por uma pessoa que desenvolve o papel de coordenador. Tarefas executadas por um grupo de pessoas são mais bem trabalhadas se houver uma pessoa fazendo a coordenação, principalmente se esse grupo for numeroso. O arcabouço possui funcionalidades que atendem diretamente os aspectos de coordenação, como as sugeridas por David (2004): participômetro e contribuímetro.

O terceiro elemento do modelo 3C é a cooperação e esse é o foco do *group storytelling*. O trabalho cooperativo de se construir a história na forma de fragmentos é a funcionalidade central do arcabouço e é função deste prover o espaço compartilhado para que os contadores da história trabalhem colaborando uns com os outros.

O modelo 3C nos ofereceu a separação conceitual que foi utilizada para se criar a modularização do arcabouço. Existem componentes que atendem aos aspectos de comunicação, enquanto outros atendem a coordenação e a cooperação. Essa componentização facilita a compreensão da estrutura.

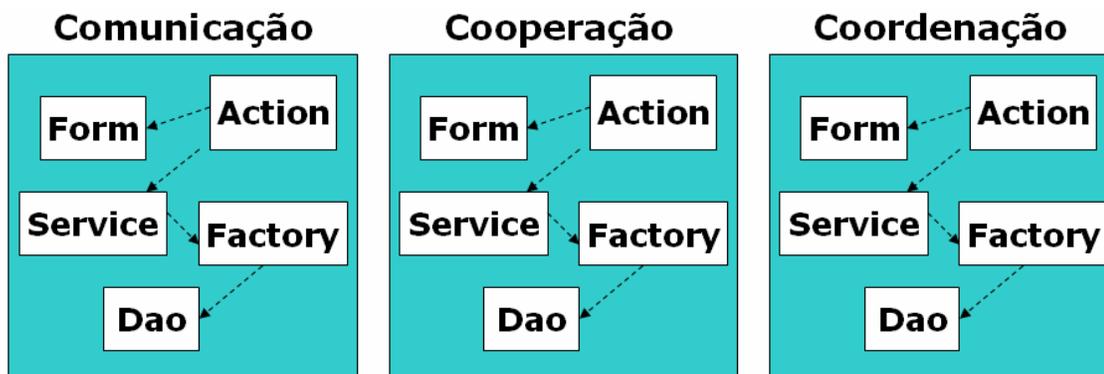


Figura 12 - Modularização baseada no modelo 3C

A Figura 12 apresenta a separação conceitual nos módulos de comunicação, cooperação e coordenação. É importante notar que cada um desses módulos contém um conjunto de classes que juntas atendem aos requisitos específicos do módulo em questão. O conjunto de classes se repete em cada um dos módulos de modo que a arquitetura de implementação e a interação entre as classes sejam padronizadas.

5.4.2 Funcionalidades

5.4.2.1 Cooperação

Funcionalidades ligadas às atividades de cooperação estão presentes em componentes que se encaixam no módulo de cooperação. Qualquer nova funcionalidade de extensão que seja implementada por um desenvolvedor que planeja instanciar o arcabouço deve entrar nesse módulo se ela estiver ligada ao trabalho em conjunto feito pelo grupo.

É nesse módulo que se encontram os componentes que atendem às regras de negócio de construção cooperativa das histórias. Boa parte do que está implementado no arcabouço fica contido aqui já que essa funcionalidade estava presente em todas as aplicações de apoio ao *group storytelling* estudadas o que a torna parte essencial do arcabouço.

Dentre as funcionalidades implementadas no módulo de cooperação podemos citar:

- **Cadastrar história** – o usuário deve informar título e descrição e deve escolher as configurações específicas da história que ele está criando. Ele também tem a opção de escolher um *template* de configuração pré-cadastrado;
- **Incluir fragmento** – o usuário deve informar título, tipo, categoria e conteúdo do fragmento. Dependendo da configuração da história para a qual ele está contribuindo, ele poderá ter a opção de se identificar ou não ao incluir um novo fragmento;
- **Incluir comentário** – o usuário informa apenas o conteúdo do comentário que passa a ficar agregado ao fragmento;
- **Associar fragmentos** – o usuário deve selecionar dois fragmentos e um tipo de associação. Além disso, ele pode incluir uma descrição para essa associação;
- **Incluir mídia** – o usuário deve informar título e descrição, além de selecionar o tipo de mídia e preencher o campo conteúdo com o trecho de código que aponta para o conteúdo remoto;
- **Incluir termo de glossário** – o usuário deve informar o nome e a descrição do novo termo de glossário;

5.4.2.2 Coordenação

Aqui entram componentes que implementam funcionalidades que atendem aos requisitos de coordenação. Serviços que ajudam na gestão dos participantes da dinâmica e de suas atividades se enquadram nessa categoria. O arcabouço contém dois serviços que se encaixam nessa categoria: o participômetro e o contribuitômetro.

O contribuitômetro é uma tabela que apresenta o total de fragmentos, comentários e mídias que um contador incluiu numa história. Através desses números o facilitador pode monitorar a quantidade de contribuições de cada contador da história.

O participômetro é outro instrumento que ajuda na coordenação da história, pois através dele o facilitador pode saber se os contadores não só estão contribuindo como também estão acessando as contribuições dos outros participantes.

5.4.2.3 Comunicação

O quadro de avisos é a única funcionalidade de comunicação implementada no arcabouço. Outras funcionalidades não foram desenvolvidas pelos possíveis problemas que podem acontecer, problemas estes que apresentamos na seção anterior, e também pela limitação de tempo para elaboração da dissertação. O quadro de avisos permite que o facilitador de uma história envie recados para os contadores. Esses recados são lidos pela primeira vez quando o contador acessa o sistema e escolhe a história em questão. Posteriormente, os recados podem ser lidos quando o contador acessa o quadro de avisos.

Apesar de não terem sido implementadas outras funcionalidades de comunicação, o arcabouço prevê, como opção de configuração de história, a permissão ou não para o uso de recursos extras de comunicação. Essa flexibilidade foi implementada para ilustrar as possibilidades de configuração por parte do facilitador.

5.5 Flexibilidade e variabilidade oferecidas pelo arcabouço

O arcabouço Sofia é um sistema semi-completo, que deve ser utilizado para criação de sistemas de apoio ao *group storytelling*. Por ser um arcabouço, ele deve oferecer flexibilidades para aqueles que irão instanciá-lo. As flexibilidades oferecidas podem ser classificadas em 3 níveis distintos e para cada um destes níveis podemos associar um papel de usuário do arcabouço.

O nível 1, que chamamos de nível *hard*, é aquele que envolve qualquer tipo de alteração no código do arcabouço. Inclusão de novas funcionalidades, alteração no modelo de

dados ou alterações na arquitetura são exemplos de situações que se encaixam neste nível. Esse tipo de flexibilidade é utilizada pelo usuário com perfil de desenvolvedor, que pretende instanciar o arcabouço fazendo modificações no mesmo.

O nível 2, ou nível de instância do sistema, envolve alterações que afetam apenas uma instalação do arcabouço. Essas alterações não implicam em mudança de código e são feitas por funcionalidades que já estão implementadas. Como exemplo de alterações desse tipo podemos citar: inclusão de novos tipos e categorias default de fragmentos, inclusão de novos tipos default de associação ou, inclusão de novos tipos default de mídia. Esse tipo de flexibilidade é utilizada por usuários com perfil de administrador do sistema. Elas são oferecidas através de uma interface de administração.

O nível 3, ou nível de histórias, envolve alterações ou configurações que afetam apenas uma história do sistema. As configurações feitas em uma história se aplicam apenas a esta história e estão isoladas do resto do sistema. Esse tipo de flexibilidade atende o usuário com papel de facilitador, ou seja, aquele que coordena a dinâmica de construção de uma história.

Tabela 6 - Quadro com níveis de flexibilidade oferecidas pelo arcabouço

Nível	Nome	Descrição	Responsável	Impacto
1	Nível Hard	Alterações no código	Desenvolvedor	Afeta todas as instâncias feitas após as alterações
2	Nível de Instância do Sistema	Alterações administrativas	Administrador	Afeta apenas a instância que está sendo alterada
3	Nível de Histórias	Configurações ou alterações de uma história	Facilitador	Afeta apenas a história em questão

No momento de criação de uma nova história, além de informar dados como título e descrição, o facilitador deve escolher algumas opções de configuração. A alternativa em relação a escolher essas opções é selecionar um template pré-configurado no sistema. Esses

templates são úteis para quando o facilitador quer criar uma história para aplicar em uma situação prevista. Dentre as opções de configuração disponíveis encontram-se:

- se a história é pública ou privada;
- se a história terá contribuições anônimas, identificadas ou se essa escolha será do contador;
- se a história não usará anexos (mídias ou outros artefatos como documentos, planilhas, etc), se usará apenas anexos por upload, se apenas anexos por mashup (anexos distribuídos), ou se ambas as modalidades;
- se a história fará uso de recursos adicionais de comunicação entre os participantes;
- se os contadores da história terão acesso às informações do contribuído e do participômetro;
- incluir ou excluir tipos e categorias de fragmentos (originalmente os default estarão disponíveis);
- incluir ou excluir novos tipos de associações entre fragmentos (originalmente os default estarão disponíveis);
- incluir ou excluir novos tipos de mídia (originalmente os default estarão disponíveis).

Para configurar cada uma dessas opções à vontade, o facilitador deve optar por uma história customizada. Caso contrário, ele deverá selecionar algum dos *templates* pré-configurados como os apresentados na Tabela 7 (os *templates* usam tipos e categorias *default* de fragmentos, associações e mídias).

Tabela 7 - Templates pré-estabelecidos para configuração de histórias

Template	Descrição
História privada	Dinâmica em que a história não está disponível para todos e

identificada	cujo conteúdo tem um grau mais elevado de confidencialidade. Os participantes são todos convidados pelo facilitador. As contribuições são necessariamente identificadas. Permite a inclusão de anexos, mas apenas por <i>upload</i>. Não faz uso de recursos adicionais de comunicação.
História privada com anonimato liberado	Template semelhante ao anterior, porém com possibilidade de contribuições anônimas. Aplicado nas situações em que as pessoas podem achar que vão ser prejudicadas pelo que relatarem. Permite que seja uma escolha do contador o anonimato ou não.
História restrita	História privada, com possibilidade de contribuições anônimas, com anexos tanto por <i>upload</i> como por <i>mashup</i> e com recursos de comunicação adicionais disponíveis.
História pública controlada	História com contribuições necessariamente identificadas, anexos por <i>upload</i> e sem uso de recursos adicionais de comunicação.
História aberta ao público	História pública com contribuições identificadas. Usa anexos apenas por <i>mashup</i> e recursos adicionais de comunicação.
História pública auto gerenciada	Tem as mesmas configurações da "História aberta ao público" exceto pelo fato dos contadores terem acesso às informações do contribuídômetro e do participômetro. A idéia é que ocorra pouca intervenção do facilitador e os próprios contadores possam conduzir o processo da dinâmica.

5.6 Classificação do arcabouço desta dissertação

No capítulo 4 apresentamos dois critérios de classificação para arcabouços. Um deles nos ajuda a entender a maneira como o arcabouço é utilizado e o outro identifica onde o arcabouço é aplicado.

Quanto à técnica de utilização, que distingue arcabouços em caixa-branca ou caixa-preta, aceitando tons de cinza dentre esses dois extremos, podemos dizer que o arcabouço

dessa dissertação é um caixa-cinza. Ele tende para caixa-branca na medida em que exige que o desenvolvedor saiba como é o seu funcionamento interno para que possa utilizá-lo e estendê-lo, mas “escurece” quando adota como uma de suas características o encaixe de componentes.

Quanto ao escopo de utilização o nosso arcabouço é claramente um arcabouço de aplicação, cujo domínio é a área de gestão do conhecimento, mais especificamente para apoio ao *group storytelling*.

6 Sofia - Implementação

Neste capítulo apresentaremos os detalhes de implementação do arcabouço. Conforme descrito no capítulo anterior, o arcabouço proposto é um software parcialmente completo, cuja instanciação é feita pela extensão ou pela complementação das funcionalidades já desenvolvidas. Ele foi construído para ser instanciado como uma aplicação Web. Apresentaremos a arquitetura do arcabouço assim como detalhes de seus componentes para que os indivíduos que forem utilizá-lo o façam compreendendo a estrutura de implementação. No final do capítulo apresentamos um passo a passo de instanciação do arcabouço.

6.1 Alicerces de construção do arcabouço

Alguns aspectos estão tão fortemente presentes na arquitetura do arcabouço que podemos denominá-los alicerces de construção. São diretrizes que surgiram pelo estudo do domínio, pelo estudo das metodologias de construção de arcabouços e também pelo conhecimento das boas práticas de desenvolvimento de sistemas. São eles:

- *Groupware* baseado no modelo 3C;
- Desenvolvimento baseado em componentes;
- Padrões de projeto;
- *Mashup*;
- Código aberto.

Apresentamos alguns detalhes do modelo 3C no capítulo 5, para que pudéssemos classificar as funcionalidades presentes no arcabouço segundo esse modelo. Agora vamos apresentar alguns detalhes de implementação do arcabouço focando em DBC, padrões de projeto e *mashup*.

Não nos aprofundaremos na questão de código aberto. É importante apenas ressaltar que é normal que trabalhos do meio acadêmico façam uso de sistemas de código aberto não só pelo custo como pela filosofia por trás dessa tendência. Os benefícios da adoção de tecnologias dessa categoria vão além da ausência de cobrança e estão enumerados e muito bem discutidos em (GBDIRECT, 2007): confiabilidade, estabilidade, auditabilidade, custo, flexibilidade e liberdade, suporte e responsabilidade. As tecnologias e as ferramentas adotadas para construção do arcabouço são todas de código aberto.

6.2 Visão geral do arcabouço

Após a apresentação das especificações do arcabouço no capítulo anterior, é necessário que entremos nos detalhes de implementação, apresentando os aspectos técnicos da solução. Um dos requisitos não funcionais que procuramos atender na elaboração do arcabouço foi procurar desenvolvê-lo de forma que fosse fácil sua compreensão e a sua utilização. A adoção de soluções já difundidas e consagradas no desenvolvimento de software busca esse objetivo.

A Figura 13 apresenta uma visão geral da arquitetura do arcabouço incluindo a separação em camadas (visão, controle e modelo, com esta última sendo ainda separada em serviço e dado), a separação conceitual baseada no modelo 3C (coordenação, cooperação e comunicação), além de outro pequeno detalhe, que é o uso do padrão *Factory* para acesso de uma camada a outra em alguns pontos.

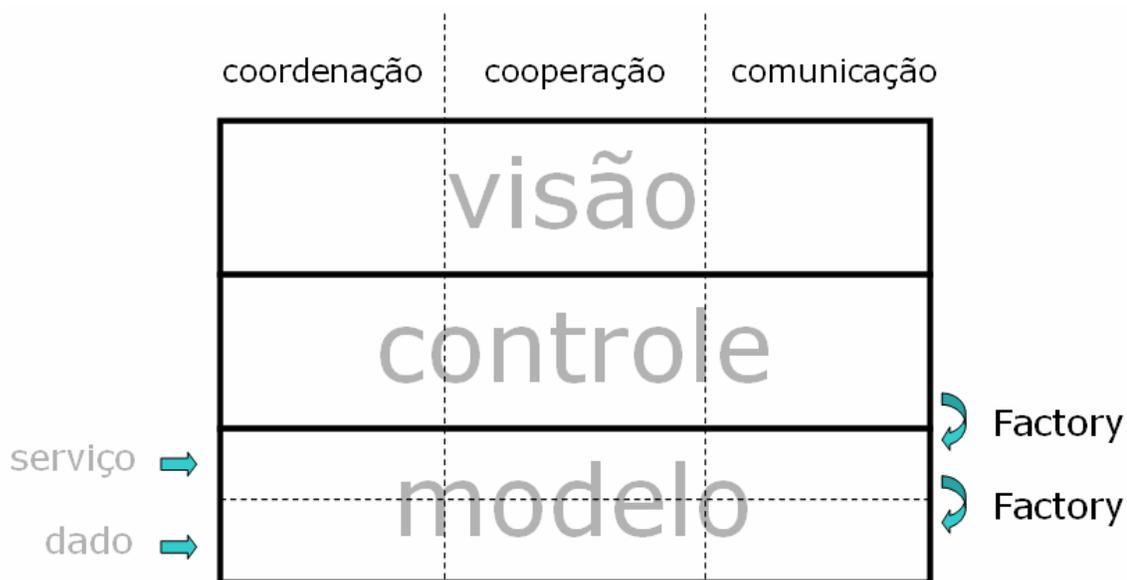


Figura 13 - Visão global da solução

Os detalhes de cada um destes pontos de observação são discutidos neste capítulo e a justificativa para a adoção deles é apresentada no decorrer do texto. Além disso, apresentaremos também outros detalhes técnicos que não aparecem nesta figura, mas que estão diretamente ligados à solução concreta tais como: linguagem de programação adotada, solução de SGBD escolhida, padrões de projeto envolvidos, aspectos envolvidos na implementação da interface com o usuário, dentre outros detalhes.

6.2 Linguagem de programação

A decisão sobre qual linguagem de programação a se utilizar no desenvolvimento de um projeto pode ser feita baseada em diferentes critérios. Alguns escolhem preocupados com o desempenho do sistema, outros pelo custo de desenvolvimento, outros pela popularidade da linguagem, outros pela linguagem ser uma tecnologia mais moderna e atualizada, e ainda, diversos outros critérios de decisão. Java (JAVA, 1995) foi a linguagem que utilizamos para implementar o arcabouço.

Essa decisão foi baseada nas características da linguagem, por ser orientada a objetos, e na popularidade que ela alcançou nos últimos anos. Uma linguagem simples, orientada a objeto, preparada para redes, interpretada, robusta, segura, arquiteturalmente neutra, portátil, de alto desempenho, *multithread* e dinâmica (JAVA, 2007).

A popularidade da linguagem pode ser comprovada por meio da observação de diferentes fatores. Welton (2005) utilizou dados de buscas no Google, valor pago pelo *click*, uso em novos projetos e ofertas de emprego para se trabalhar com a linguagem para fazer uma comparação entre elas. Labelle (2006) apresenta um gráfico de utilização de linguagens nos projetos de código aberto hospedados no site *sourceforge.net*. O'Reilly (2007) apresenta os dados sobre vendas de livros sobre linguagens de programação entre janeiro de 2006 e janeiro de 2007. Ertl (2007) tenta estimar a popularidade das linguagens de programação fazendo uma comparação entre os números de mensagens enviadas para as listas de discussão de cada uma das linguagens.

Essas fontes de informação comprovam que Java é uma linguagem bem popular e umas das mais difundidas e utilizadas atualmente. Ao adotarmos essa linguagem estamos possibilitando que um maior número de pessoas possa compreender o arcabouço e conseqüentemente, possa utilizá-lo.

6.3 Padrões

A literatura está repleta de definições para padrões de projeto. Segundo Christopher (1977), “Cada padrão é uma regra de 3 partes, que expressa uma relação entre um determinado contexto, com um problema, e uma solução”. Fowler (1997) já prefere algo mais sucinto quando afirma que “um padrão é uma idéia que foi útil num contexto prático e que provavelmente será útil em outros”. Nesse trabalho, consideramos um padrão de projeto como uma solução para um problema recorrente dentro de um contexto.

A abstração de padrão de projeto pode ser considerada tanto no nível micro, de linha de código, como no nível macro, de arquitetura da aplicação. O arcabouço que estamos propondo procura usar padrões indiferentemente da granulosidade de observação.

Todas as metodologias de construção de arcabouços estudadas sugerem a utilização de padrões de projeto. É unanimidade entre os autores que a utilização de padrões traz benefícios. Alur *et al.* (2001) citam 3 principais vantagens: garante a utilização de uma solução comprovada, fornece um vocabulário comum e restringe o espaço de solução.

Os padrões foram utilizados tanto no nível macro, com a arquitetura *Model 2*, como num nível menor, onde podemos citar os padrões *Composite View*, *Session Facade* e *Data Access Object (DAO)*, dentre outros. A seguir apresentaremos os detalhes de implementação e falaremos um pouco sobre cada um desses padrões.

6.3.1 Model 2 como macro arquitetura

O padrão *Model 2* é baseado na arquitetura MVC (BURBECK, 1992). Essa arquitetura propõe o desacoplamento dos elementos modelo, visão e controle. O modelo representa os dados e a lógica de negócio da aplicação para trabalhar com esses dados, a visão é usada para apresentar o conteúdo do modelo de diferentes maneiras e o controle transforma interações do usuário com a visão em ações a serem realizadas sobre o modelo.

No padrão *Model 2*, os *servlets* fazem o papel de controlador enquanto os *jsps* fazem o papel do elemento visão. O modelo é aqui representado por JavaBeans. Conforme apresentado na Figura 14, um cliente interage com um *servlet*, que instancia um *javabean* e repassa o fluxo para o *jsp* que utiliza o *javabean* para montar a visão e responder ao cliente.

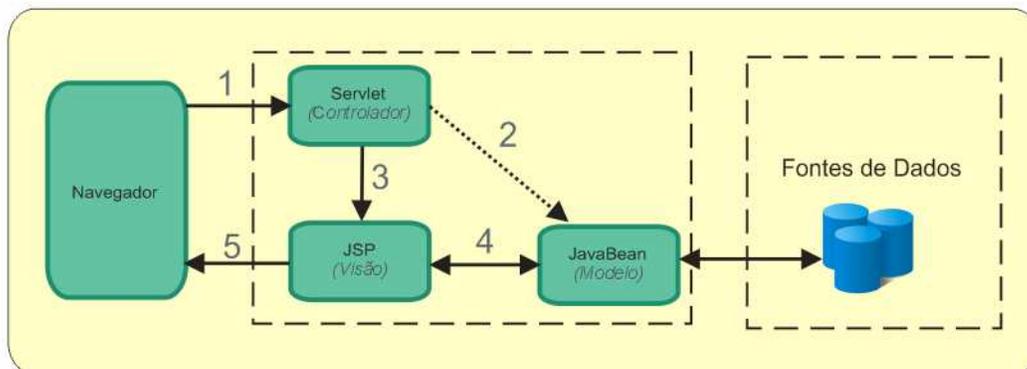


Figura 14 - Diagrama representando o padrão Model 2 - extraída de (VIADIGITAL, 2007)

Para implantar a arquitetura Model 2 no arcabouço utilizamos a tecnologia Struts (STRUTS, 2007) que está muito bem documentada na Internet além de estar madura e ser bem popular.

6.3.2 Padrões de projeto

Além da macro-arquitetura do arcabouço ser definida com um padrão bem difundido de desenvolvimento, também foram utilizados padrões de projeto num nível menor de observação. A camada de apresentação, assim como camadas de lógica de negócio e de acesso a dados, foi implementada seguindo essa filosofia.

Na camada de apresentação utilizamos o padrão *Composite View*. Esse padrão permite que façamos a montagem da tela de apresentação através da composição de partes. Um modelo de tela é definido com partes configuráveis e o encaixe dessas partes pode ser feito dinamicamente. Esse padrão permite que possamos configurar as telas com maior flexibilidade e permite que possamos reutilizar modelos de tela e partes para composição. Esse padrão foi implantado utilizando-se a tecnologia Tiles (TILES, 2007) integrada com o Struts.

Na camada onde são implementadas as regras de negócio, que chamamos de camada de serviço, utilizamos o padrão *Session Facade*. Esse padrão serve como uma fachada que

encapsula a complexidade das regras de negócio e só expõe métodos que são relevantes para o cliente do objeto de negócio.

Para criar uma camada de abstração de acesso a dados utilizamos o padrão *Data Access Object (DAO)*. Esse padrão permite que todo o acesso a dados seja encapsulado e ele controla a conexão com a fonte de dados para se obter ou se armazenar informações. Seu uso permite que, por exemplo, possamos trocar todo o meio de armazenamento de dados sem que isso impacte em toda a aplicação.

O acesso de uma camada a outra do arcabouço é feito de forma indireta com o uso do padrão *Factory*. A adoção desse padrão neste ponto permite que se possa fazer uma troca completa de uma camada do arcabouço sem que isso afete as outras camadas bastando que a nova camada implementada atenda aos requisitos das interfaces definidas. Esse padrão é utilizado no momento em que uma classe do Struts vai acessar a camada de serviço, assim como quando um componente da camada de serviço vai acessar a camada de dados.

Além desses padrões alguns outros são utilizados como infra-estrutura para o arcabouço. A tecnologia Struts que adotamos para construção do arcabouço faz uso do padrão *Front Controller*, que centraliza o processamento de requisições à aplicação. O padrão *Transfer Object* é utilizado para encapsular dados de negócio. Mais detalhes sobre esses padrões de projeto podem ser encontrados em (PATTERNS, 2007) e em (GAMMA, 1995).

6.4 Detalhes de implementação da interface

Mantendo a regra de usar padrões na elaboração do arcabouço, o desenvolvimento da interface com o usuário também foi feito desta maneira. A padronização que tem sido adotada na implementação de interfaces de sistemas *Web*, e que foi usada com o Sofia, é aquela que prega a separação em estrutura, apresentação e comportamento. A estrutura é definida pelo HTML, a apresentação pelo CSS e o comportamento pelo javascript.

Existem diversas vantagens em se adotar essa separação, mas Sutherland (2008) destaca 3 principais: redução do uso de banda, pois menos dados têm que trafegar pela rede, melhoria em relação a significado e acessibilidade do código de marcação permitindo que outros dispositivos possam acessar o mesmo site, e, facilidade para manutenção do código devido ao uso de padrões. Todos esses benefícios são interessantes para o arcabouço, mas o último é o que está mais alinhado com a idéia de facilidade de uso e compreensão que queremos garantir para o arcabouço.

Na parte relacionada a comportamento, que fica definida pelo código javascript utilizado pela aplicação, foi adotada a biblioteca Prototype (PROTOTYPE, 2008). Essa biblioteca define uma série de funções que facilitam o desenvolvimento de aplicações Web dinâmicas.

Uma característica do arcabouço que está diretamente ligada à questão da interface com o usuário é o conjunto de idiomas em que ele é apresentado. A utilização de um único idioma certamente limita o universo de possíveis usuários e para evitar essa restrição o arcabouço foi desenvolvido com o conceito de internacionalização. Todo o texto que é apresentado pelo sistema fica armazenado num arquivo que está associado a um idioma e o sistema utiliza o arquivo que corresponde ao idioma selecionado pelo usuário. Atualmente o arcabouço possui arquivos para os idiomas português e inglês. A seleção do idioma é feita na tela inicial do arcabouço.

6.5 Componentes

A abordagem de desenvolvimento baseado em componentes determina que uma aplicação seja constituída a partir de um conjunto de módulos (componentes) interligados. A modularização citada aqui se encaixa perfeitamente num dos princípios básicos das ciências em geral: dividir para conquistar.

Segundo Sametinger *apud* Almeida *et al.* [2007, p.59], “Componentes de software reutilizáveis são artefatos auto-contidos, claramente identificáveis, que descrevem e/ou executam funções específicas e têm interfaces claras, documentação apropriada e um estado de reuso definido”.

Reuso, redução de complexidade, de custo e melhoria da qualidade do software são apenas algumas das vantagens obtidas. O uso de componentes para construção do arcabouço permite que ele seja mais bem compreendido por quem irá utilizá-lo e esse é um dos pré-requisitos para que arcabouços que têm características de caixa-branca sejam aceitos e utilizados. Um desenvolvedor que quiser estender ou modificar alguma funcionalidade do arcabouço trabalhará apenas num determinado componente.

Uma das diretrizes de implementação do arcabouço é o desenvolvimento baseado em componentes. A modularização do arcabouço em componentes sendo feita baseada no modelo 3C ajuda na compreensão, e conseqüentemente, na extensão e complementação do mesmo. Os componentes foram implementados de modo que se encaixem em um dos elementos do modelo 3C: comunicação, coordenação ou cooperação. A estrutura de pacotes das classes desenvolvidas para o arcabouço obedece a essa modularização e as classes que atendem a funcionalidades ligadas a cada um desses elementos devem utilizar o pacote correspondente. A Tabela 8 apresenta esses valores.

Tabela 8 - Estrutura de pacotes do arcabouço de acordo com o modelo 3C

<code>br.ufrj.ppgi.sofia.comunicacao.*</code>
<code>br.ufrj.ppgi.sofia.coordenacao.*</code>
<code>br.ufrj.ppgi.sofia.cooperacao.*</code>

Para implementação dos componentes que atendem aos requisitos de negócio do arcabouço inicialmente pensamos em adotar a tecnologia *Enterprise JavaBeans* (EJB, 2007) mas depois decidimos utilizar classes java planas, chamadas de POJOs. Essa decisão foi feita

com o intuito de tornar mais simples a arquitetura do arcabouço. Para garantir o controle de transação das operações executadas com estas classes foi utilizada a tecnologia AOP.

6.5.1 Controle de transação com AOP

O controle de transação feito com AOP que utilizamos no arcabouço não exige do desenvolvedor conhecimento relacionado a essa tecnologia. A demarcação das transações é feita com uma anotação logo acima da declaração de um método da classe e essa demarcação é feita com os valores: *Required*, *RequiresNew*, *Supports*, *NotSupported* ou *Never*. Na implementação do arcabouço foram utilizados basicamente os valores *Required* e *Supports* com o primeiro sendo utilizado em métodos que fazem alteração no banco de dados e o segundo nos métodos que apenas fazem consultas ao banco. A Tabela 9 exemplifica esse tipo de declaração.

Tabela 9 - Controle de transação com AOP

<code>@Tx(TxType.SUPPORTS)</code> <code>public Historia consultarHistoria(...</code>
<code>@Tx(TxType.REQUIRED)</code> <code>public void cadastrarHistoria(Historia historia) ...</code>

6.6 Mashups

Os primeiros pesquisadores que pensaram no conceito de componentes imaginaram que sistemas pudessem ser desenvolvidos por meio da combinação desses elementos exigindo pouco esforço de quem estivesse fazendo essa montagem. Nos últimos anos, estamos vendo aplicações surgirem na Internet usando exatamente esse conceito. Um *mashup* é uma aplicação *Web* que combina dados de mais de uma fonte em uma única ferramenta integrada (MASHUP, 2007).

Mashup é um dos elementos que formam a chamada *Web 2.0* (WEB2.0, 2004), que consiste numa gama de comunidades e serviços hospedados na Internet que facilitam a

colaboração e o compartilhamento entre usuário. Redes sociais e Wikis são exemplos de aplicações dessa categoria.

Uma das características que resolvemos aplicar ao arcabouço foi o conceito de *mashup*. Fizemos isso para permitir que novas funcionalidades fossem agregadas facilmente sem que fosse necessário grande esforço por parte do desenvolvedor. Foi através dessa flexibilidade do arcabouço que pudemos agregar a utilização de outras mídias como som, áudio e vídeo, dentre outros recursos de auxílio ao processo de *group storytelling*. Além dessa vantagem, consideramos que o uso de *mashup* acrescenta um fator motivacional ao processo e estimula a utilização do conceito por desenvolvedores que forem instanciar o arcabouço.

Os trabalhos anteriores que implementaram soluções de apoio ao *group storytelling* citaram a importância da utilização de outras mídias para complementação da dinâmica, porém não utilizaram esses artefatos em suas implementações. O arcabouço proposto nessa dissertação agrega o uso de outras mídias utilizando o conceito de *mashup* fazendo uso do serviço www.youtube.com para vídeos, picasaweb.google.com.br para fotos e, media.imeem.com para áudio.

A utilização desses serviços para apoio à dinâmica de *group storytelling* será um fator motivacional de participação visto o grande sucesso desses serviços na Internet atualmente. Além disso, o armazenamento dessas mídias em servidores diferentes daquele onde o arcabouço está instanciado poupa espaço em disco.

A implantação do conceito de *mashup* no arcabouço faz com que desenvolvedores que planejem instanciá-lo estendam o mesmo com outros serviços diferentes daqueles já utilizados. Todos os dias surgem novos serviços que podem ser somados às aplicações. Mapas de localização (maps.google.com), calendários (www.google.com/calendar) e tabelas (spreadsheets.google.com) são apenas alguns dos serviços que poderiam ser facilmente agregados.

6.7 Interação entre as classes

No capítulo anterior, apresentamos a modularização baseada no modelo 3C com cada módulo englobando um conjunto de classes que interagem entre si para realizar os requisitos específicos daquele módulo. Naquele momento não foram apresentados os detalhes de interação entre as classes nem tampouco o conjunto completo que atendia a alguma funcionalidade específica.

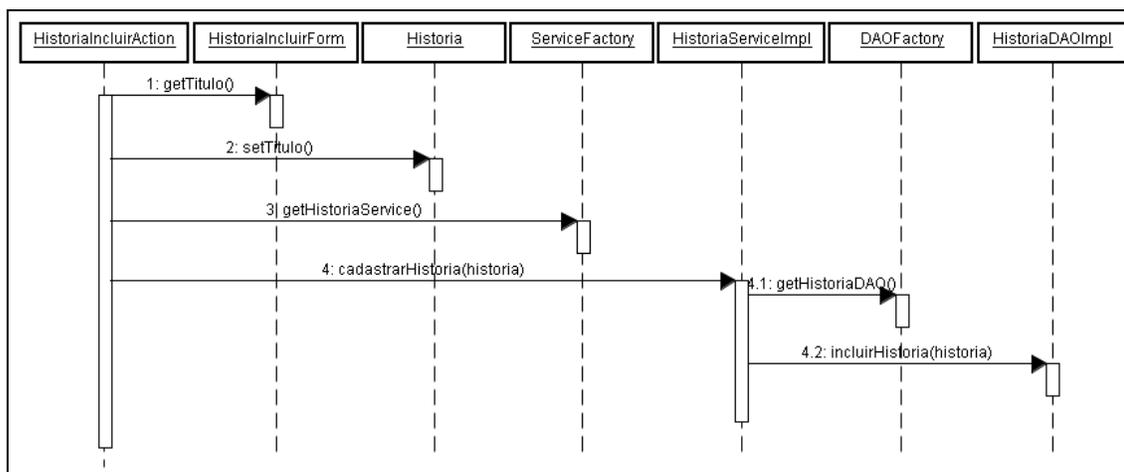


Figura 15 - Diagrama de seqüência da funcionalidade Cadastrar História

A Figura 15 contém o diagrama de seqüência que apresenta a interação entre as classes de implementação que realizam a funcionalidade de cadastro de uma nova história. Essa arquitetura de implementação se repete em todas as funcionalidades implementadas no arcabouço Sofia. O diagrama não contém todas as chamadas de método, mas procura apresentar as classes envolvidas na funcionalidade desde o início da requisição até a camada que acessa o banco de dados.

6.8 Modelo de dados

O modelo de dados apresentado na Figura 16 exhibe as entidades e os relacionamentos entre elas, porém, devido ao pouco espaço, esse modelo não apresenta os atributos e algumas

tabelas que consideramos como secundárias. No anexo 2 é apresentado o modelo completo. O modelo deve ser usado para o entendimento das entidades envolvidas na implementação do arcabouço.

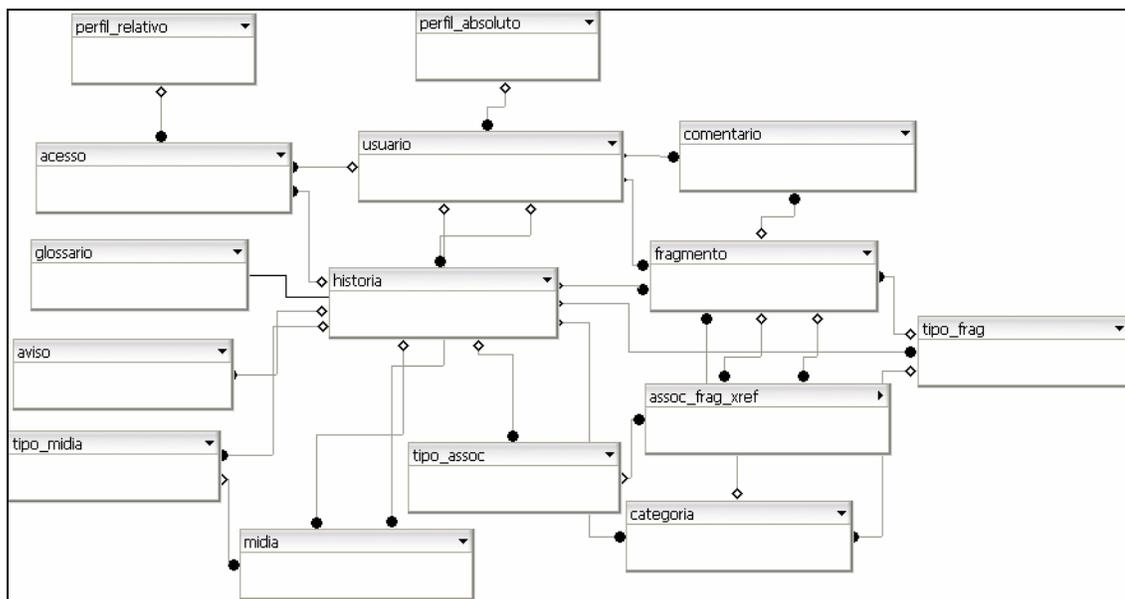


Figura 16 - Modelo de dados do arcabouço

Para uma melhor compreensão do modelo de dados, podemos fazer um paralelo entre as funcionalidades presentes no arcabouço e as entidades apresentadas na Figura 16. A Tabela 10 apresenta um quadro com essa correlação.

Tabela 10 - Relação entre funcionalidades e entidades do modelo de dados

Funcionalidade	Entidades relacionadas
Cadastrar História	O cadastro de histórias está relacionado com as entidades historia , usuario , acesso e perfil_relativo . O usuário que cadastrou a história passa a ter acesso de facilitador aquela história.
Incluir Fragmento	A inclusão de fragmento está relacionada às entidades fragmento , tipo_frag , categoria ,

	<p>historia e usuario. Ao incluir um fragmento em uma história, o autor daquele fragmento deve escolher um tipo e uma categoria para classificar o fragmento.</p>
Incluir Comentário	<p>A inclusão de comentários está relacionada às entidades comentario, fragmento e usuario. Os comentários têm autor e estão associados a um fragmento.</p>
Associar Fragmentos	<p>A associação entre fragmentos está relacionada às entidades fragmento, assoc_frag_xref e tipo_assoc. Um usuário pode associar um fragmento com outro fragmento selecionando um tipo para classificar essa associação.</p>
Incluir Mídia	<p>A inclusão de mídias está associada às entidades midia, tipo_midia e historia. Os usuários podem incluir mídias associando-as a uma história e eles devem classificar essa mídia selecionando um tipo.</p>
Incluir Termo de Glossário	<p>A inclusão de termos de glossário está relacionada às entidades glossario e historia. Os contadores de uma história podem incluir termos de glossário associando-os a história.</p>
Contribuômetro	<p>As entidades fragmento, comentario, midia e usuario estão relacionadas a essa</p>

	funcionalidade. O contribuímetro apresenta a quantidade de cada um desses itens por contador da historia.
Participômetro	As entidades fragmento , comentario , midia e usuario , além de outras que são apresentadas apenas no anexo, estão relacionadas a essa funcionalidade. Nessa funcionalidade é preciso que seja medida a quantidade de acessos dos contadores às contribuições dos outros participantes.
Quadro de avisos	O quadro de avisos está relacionado às entidades aviso e historia . O facilitador de uma história pode cadastrar avisos que são lidos pelos contadores.

6.9 Ambiente de instanciação

O arcabouço, como aplicação semi-completa, exige alguns serviços de infra-estrutura para que possa ser instanciado. O ambiente de instanciação deve conter uma instalação de MySQL 5.0 e uma instalação de JBoss 4.2. Esses são os requisitos básicos para que se possa instalar e configurar o arcabouço. Os detalhes de instalação e de configuração de todos os serviços estão descritos no Anexo 1 – Preparação do ambiente. O serviço de e-mail depende de um servidor SMTP, mas atualmente o arcabouço está configurado para utilizar um servidor externo ao ambiente de instalação. A Figura 17 apresenta um diagrama do ambiente.

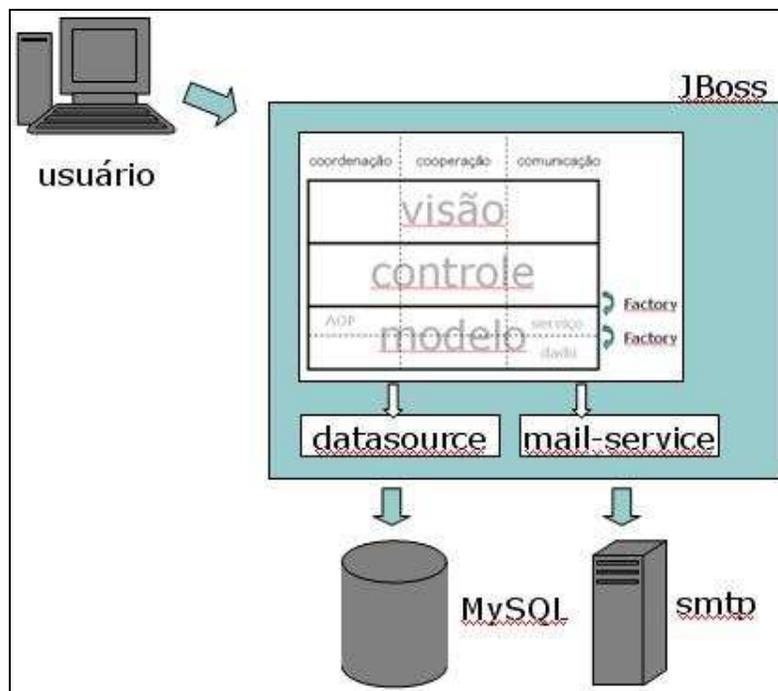


Figura 17 - Ambiente de instanciação do arcabouço

6.10 Passo-a-passo para instanciação do arcabouço

O passo-a-passo de instanciação do arcabouço serve para deixar claro o que deve ser feito quando alguém decide usá-lo para criar uma aplicação de apoio ao *group storytelling*. Algumas perguntas devem ser respondidas e de acordo com as respostas para estas perguntas algumas ações devem ser tomadas.

O arcabouço, do modo como está implementado atualmente, já atende a todos os requisitos da aplicação que se quer desenvolver?

Se a resposta para essa pergunta for sim, então a instanciação deve ser feita sem que nada seja modificado. Entretanto, se a resposta for não, devem ser definidas as partes do arcabouço que sofrerão modificações. As modificações podem ser separadas em dois grandes grupos: o primeiro deles inclui aquelas modificações horizontais, que alteram uma camada inteira do arcabouço sem que isso afete as outras camadas. Alterar toda a interface com o usuário, alterar o SGBD ou alterar a camada de serviço para usar outra tecnologia de

componentes seriam modificações que se encaixariam neste grupo. O segundo grupo inclui aquelas modificações verticais, que implicam em pequenas alterações em cada uma das camadas. A inclusão de novas funcionalidades se encaixa neste grupo. Novas funcionalidades implicam em desenvolvimento de novas telas, novos componentes de negócio, possivelmente alteração no modelo de dados, dentre outras modificações. Independente do tipo de modificação que se queira fazer uma das regras que devem ser mantidas é o uso do modelo 3C para separação das funcionalidades.

A modificação que se pretende fazer é uma modificação horizontal ou vertical?

Se for horizontal, ela deve ser feita com a nova camada mantendo a separação descrita pelo modelo 3C e mantendo os padrões de implementação adotados no arcabouço. Se for vertical, ela também deve manter a separação descrita pelo modelo 3C e os padrões de implementação, mas novas perguntas devem ser respondidas.

A modificação implica em alteração no modelo de dados?

Se sim, as modificações devem ser feitas e os scripts de preparação da base de dados devem ser atualizados. Um software que auxilia na alteração do modelo de dados e no sincronismo dessas modificações com a instância do banco é o DBDesigner (DBDESIGNER, 2008).

A alteração implica em codificação de novos métodos de negócio?

Se sim, os novos métodos devem ser incluídos seguindo o modelo de implementação já usado, não esquecendo da declaração de controle de transação feita com AOP.

A alteração implica em criação de novas telas?

Se sim, o modelo que define a separação entre estrutura (HTML), apresentação (CSS) e comportamento (javascript) deve ser mantido. Notem que alterações podem ser feitas apenas em um destes elementos.

Não necessariamente as modificações têm que ser verticais ou horizontais, elas podem misturar características dos dois tipos, mas normalmente apenas um tipo será usado. Esse passo-a-passo define algumas regras básicas de instanciação, mas nada impede que um desenvolvedor altere o arcabouço da maneira que quiser. A manutenção dos padrões e dos modelos adotados pelo arcabouço ajudará na compreensão e no uso em futuros trabalhos. A Figura 18 apresenta um fluxograma que representa o passo-a-passo descrito nesta seção.

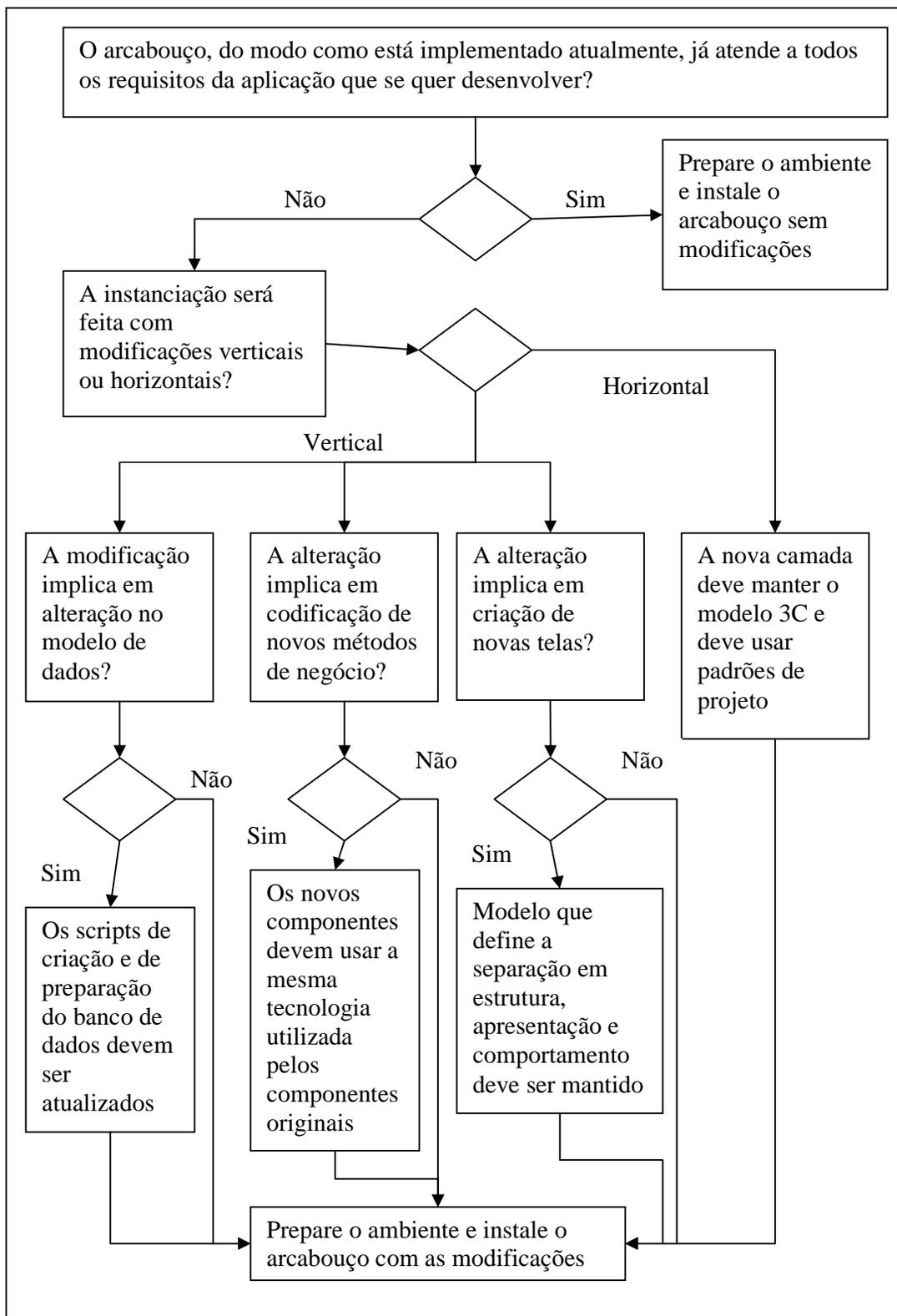


Figura 18 - Fluxograma de decisão para instanciação do arcabouço

7 Avaliação do arcabouço Sofia

Neste capítulo apresentaremos uma primeira avaliação do arcabouço proposto neste trabalho. Serão detalhadas duas abordagens: uma objetiva, baseada em métricas de software, e outra subjetiva, em que é apresentado um relatório de *feedback* escrito por um avaliador que instanciou o arcabouço.

7.1 Abordagem de avaliação

A avaliação de um arcabouço é idealmente feita através de inúmeras instâncias executadas por diferentes indivíduos. Esse é o processo natural que acontece com arcabouços em geral e com aqueles disponíveis na Internet. A própria quantidade de vezes que um arcabouço é utilizado já é um indicativo de que ele está sendo bem ou mal avaliado.

Entretanto, no escopo de um trabalho científico, uma avaliação se dá normalmente pela experimentação. Experimentação oferece um modo sistemático, disciplinado, computável e controlado para avaliação. Os objetivos relacionados à execução de experimentos em Engenharia de Software são a caracterização, avaliação, previsão, controle e melhoria a respeito de produtos, processos, recursos, modelos, teorias entre outros (TRAVASSOS; GUROV; AMARAL; 2002).

No entanto, uma avaliação formal e completa, baseada em experimentação, está fora do escopo dessa dissertação. O objetivo da nossa avaliação é encontrar indícios sobre o uso do arcabouço, apontando vantagens e dificuldades que surgem com a sua adoção. Com isso, decidimos utilizar duas abordagens de avaliação para termos um *feedback* quantitativo e/ou qualitativo do trabalho proposto.

As duas abordagens consistem na instanciação do arcabouço com a inclusão de novas funcionalidades. Na primeira instanciação são observadas algumas métricas de software que indicam, mesmo que de forma relativa, o esforço necessário para se desenvolver um sistema

usando o arcabouço como base. Na segunda instanciação, as mesmas funcionalidades são incluídas, mas desta vez o *feedback* é obtido através de depoimento do instanciador, o que denota um grau de subjetividade na avaliação.

O trabalho de avaliação consistiu na instanciação do arcabouço com a inclusão de algumas funcionalidades que estavam presentes nos sistemas propostos pelos outros estudos de *group storytelling*, mas que não estavam presentes no arcabouço. As funcionalidades agregadas foram:

- O criador de um fragmento pode editá-lo;
- O facilitador pode unir dois fragmentos em um;
- O facilitador pode separar um fragmento em dois;
- Os contadores podem fazer *upload* de arquivos anexando-os a uma história;
- Qualquer pessoa que acessa o sistema pode criar uma conta.

Essas funcionalidades foram escolhidas por que agregariam valor ao arcabouço e também por que apresentam uma complexidade mediana de codificação. Não queríamos algo muito simples que demandasse pouco esforço, nem algo muito complexo que exigisse bastante tempo para finalização da avaliação. Funcionalidades com essas características provavelmente distorceriam os nossos objetivos de avaliação.

Nas próximas seções apresentaremos os detalhes de cada uma das abordagens de avaliação executadas.

7.2 Primeira abordagem

O objetivo desta abordagem de avaliação é tentar medir por meio de métricas de software o esforço necessário para se desenvolver um sistema de *group storytelling* com o uso do arcabouço Sofia. Para efeito de comparação, foram tiradas medidas no arcabouço intacto, sem modificações, e na instanciação feita com a inclusão das funcionalidades apresentadas na seção anterior. Essas medidas foram feitas com o auxílio do programa Metrics (METRICS,

2008). Este software, por sua vez, foi fundamentado nos conceitos de métricas apresentados por (Henderson-sellers, 1996) e (Martin, 1995). A Figura 19 apresenta o relatório gerado pelo Metrics quando executado sobre o arcabouço intacto, sem modificações. A Figura 20 apresenta o mesmo relatório, mas desta vez gerado sobre uma instanciação do arcabouço. Os relatórios estão apresentados em inglês, pois este é o único idioma com o qual o software opera.

Metric	Total	Mean	Std. D...	Maximum	Resource causing Maximum
⊕ Number of Overridden Methods (avg/max per type)	28	0,235	0,604	2	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cor
⊕ Number of Attributes (avg/max per type)	142	1,193	2,44	14	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Number of Children (avg/max per type)	81	0,681	5,024	53	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Number of Classes (avg/max per packageFragment)	119	6,263	9,066	40	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cod
⊕ Method Lines of Code (avg/max per method)	3956	6,671	12,074	67	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cod
⊕ Number of Methods (avg/max per type)	583	4,899	8,611	66	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cod
⊕ Nested Block Depth (avg/max per method)		1,32	0,671	4	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cod
⊕ Depth of Inheritance Tree (avg/max per type)		2,975	1,286	4	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cor
⊕ Number of Packages	19				
⊕ Afferent Coupling (avg/max per packageFragment)		11,263	16,817	52	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Number of Interfaces (avg/max per packageFragment)	7	0,368	0,581	2	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ McCabe Cyclomatic Complexity (avg/max per method)		1,477	1,248	15	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cod
⊕ Total Lines of Code	7385				
⊕ Instability (avg/max per packageFragment)		0,502	0,345	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cor
⊕ Number of Parameters (avg/max per method)		1,115	1,189	4	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cor
⊕ Lack of Cohesion of Methods (avg/max per type)		0,186	0,326	0,959	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Efferent Coupling (avg/max per packageFragment)		5,158	8,762	40	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cod
⊕ Number of Static Methods (avg/max per type)	10	0,084	0,277	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cor
⊕ Normalized Distance (avg/max per packageFragment)		0,39	0,346	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Abstractness (avg/max per packageFragment)		0,161	0,202	0,5	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cor
⊕ Specialization Index (avg/max per type)		0,09	0,259	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cor
⊕ Weighted methods per Class (avg/max per type)	876	7,361	18,337	182	/Sofia/src/br/ufrrj/nce/ppgi/sofia/cod
⊕ Number of Static Attributes (avg/max per type)	71	0,597	2,931	32	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo

Figura 19 - Relatório de métricas do arcabouço puro

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum
⊕ Number of Overridden Methods (avg/max per type)	40	0,288	0,67	2	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Number of Attributes (avg/max per type)	167	1,201	2,36	14	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Number of Children (avg/max per type)	101	0,727	5,55	62	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Number of Classes (avg/max per packageFragment)	139	7,316	11,003	49	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Method Lines of Code (avg/max per method)	4503	6,642	11,869	67	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Number of Methods (avg/max per type)	668	4,806	8,394	69	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Nested Block Depth (avg/max per method)		1,316	0,657	4	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Depth of Inheritance Tree (avg/max per type)		3,065	1,225	4	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Number of Packages	19				
⊕ Affherent Coupling (avg/max per packageFragment)		13,632	19,863	60	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Number of Interfaces (avg/max per packageFragment)	7	0,368	0,581	2	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ McCabe Cyclomatic Complexity (avg/max per method)		1,546	1,45	15	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Total Lines of Code	8472				
⊕ Instability (avg/max per packageFragment)		0,499	0,347	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Number of Parameters (avg/max per method)		1,143	1,2	4	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Lack of Cohesion of Methods (avg/max per type)		0,189	0,325	0,959	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Efferent Coupling (avg/max per packageFragment)		6,158	10,859	49	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Number of Static Methods (avg/max per type)	10	0,072	0,258	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Normalized Distance (avg/max per packageFragment)		0,398	0,35	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo
⊕ Abstractness (avg/max per packageFragment)		0,155	0,199	0,5	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Specialization Index (avg/max per type)		0,107	0,274	1	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Weighted methods per Class (avg/max per type)	1048	7,54	18,177	189	/Sofia/src/br/ufrrj/nce/ppgi/sofia/col
⊕ Number of Static Attributes (avg/max per type)	82	0,59	2,886	34	/Sofia/src/br/ufrrj/nce/ppgi/sofia/glo

Figura 20 - Relatório de métricas da instanciação do arcabouço

Esses relatórios apresentam uma variedade de métricas, mas para nossa avaliação consideramos apenas aquelas que indicam quantidade ou organização do código. A Tabela 11 apresenta algumas destas métricas e a diferença percentual que foi percebida após a instanciação.

Tabela 11 - Quadro com métricas e aumento percentual

Métrica	Arcabouço puro	Arcabouço instanciado	Aumento percentual
Número de atributos	142	167	18%
Número de classes	119	139	17%
Número de métodos	583	668	15%
Número de pacotes	19	19	0%
Número de interfaces	7	7	0%
Total de linhas de código	7385	8472	15%
Número de métodos estáticos	10	10	0%
Número de atributos estáticos	71	82	15%

Observando o quadro podemos separar essas métricas em dois grupos distintos: aquelas que tiveram aumento percentual na faixa entre 15% e 18% e aquelas que não tiveram

aumento percentual, ficando com 0%. Se observarmos as características de cada um destes grupos podemos notar uma diferença marcante entre eles que explica o porquê desta distinção.

O primeiro grupo, que agrega atributos, classes, métodos, linhas de código e atributos estáticos se caracteriza por estar ligado ao código que define o ‘como fazer’. Já o segundo grupo, que agrega pacotes, interfaces e métodos estáticos, está ligado ao código que define o ‘o que fazer’. Estes são conceitos nativos da orientação a objeto, e é o que diferencia classes de interfaces.

Os pacotes são exatamente os elementos que definem a arquitetura do software, com a separação em camadas e a separação entre os três elementos do modelo 3C. A manutenção da quantidade de pacotes após a instanciação indica que não houve alteração em relação a esses aspectos, ou seja, a arquitetura se manteve intacta. As interfaces são os elementos responsáveis por definir comportamento e a manutenção do número de interfaces indica também que não houve alteração em relação a esse aspecto. Os métodos estáticos normalmente estão ligados a códigos chamados de utilitários que atendem a necessidades de qualquer camada do software. Não houve inclusão de novos métodos nesta instanciação.

A faixa de aumento percentual percebida no primeiro grupo de métricas não garante que qualquer tipo de instanciação do arcabouço vai exigir essa mesma quantidade de esforço. Esses números vão depender diretamente da quantidade e da complexidade de novas funcionalidades que se quer incluir numa instanciação. Em contrapartida, a manutenção ou a pouca alteração dos números relacionados ao segundo grupo de métricas é um forte indicativo de que não houve alteração em relação à arquitetura do arcabouço proposto neste trabalho.

Não podemos afirmar que o aumento percentual percebido no primeiro grupo de métricas era esperado com os valores que foram apresentados. O que podemos afirmar é que certamente haveria um aumento percentual, e que esse aumento estaria diretamente

relacionado à quantidade e complexidade das funcionalidades incluídas. Entretanto, podemos afirmar que a manutenção dos valores do segundo grupo de métricas era esperada, já que a inclusão das novas funcionalidades não implicava em alteração de arquitetura do arcabouço.

7.3 Segunda abordagem

O objetivo desta abordagem é tentar avaliar, mesmo que de forma subjetiva, as facilidades e dificuldades oferecidas pelo arcabouço. Foi solicitado que um membro do grupo de pesquisa Greco (GRECO, 2008) fizesse a instanciação do arcabouço com as funcionalidades apresentadas na seção 7.1 e que redigisse um relatório na forma de um texto livre, onde ele deveria descrever tudo que achasse relevante em termos de avaliação do arcabouço. O relatório de *feedback* é apresentado por completo na seção 7.3.1. Na seção 7.3.2 apresentamos nossas considerações sobre este relatório.

O avaliador tem como linha de estudo a área de *group storytelling*, mas não estava envolvido com o arcabouço Sofia até participar desta avaliação. Além disso, ele tem um perfil de desenvolvedor de software o que trouxe facilidades para compreensão das tecnologias utilizadas. O avaliador não recebeu treinamento, sendo introduzido ao arcabouço apenas com uma conversa informal de cerca de 45 minutos onde lhe foi apresentado o código-fonte da implementação. Ao final desta apresentação ele recebeu um CD que continha as instruções de avaliação, o código-fonte do arcabouço, e a dissertação, na versão da época, que já continha o anexo 1, onde é apresentado o passo-a-passo de preparação do ambiente de instanciação.

7.3.1 Relatório

Relatório de Avaliação – Arcabouço Sofia

Durante a implementação das funcionalidades pedidas no experimento, pude observar diversos aspectos a respeito deste arcabouço, os quais cito abaixo:

1. Reuso

Este arcabouço não é uma biblioteca de componentes "soltos" e sim uma aplicação genérica a qual pode ser estendida, o que faz com que se perca um pouco da flexibilidade na implementação de uma instância mais específica. Porém, o mesmo pode ser reutilizado, já que é genérico, agilizando o processo de construção dessa instância.

2. Aderência ao padrão MVC e separação das camadas

O padrão MVC divide a responsabilidade entre os objetos de modo que manutenções e alterações futuras no código sejam mais fáceis de serem feitas.

A divisão em camadas incide, principalmente, na robustez da aplicação, onde alterações em uma das camadas não interferirá nas demais.

No framework proposto há uma clara divisão de responsabilidades, tanto em relação ao padrão MVC, quanto em relação às camadas. As últimas estão divididas, basicamente, em:

- Apresentação: Onde reside toda a lógica de apresentação (JSP, Prototype e Struts)

- Negócio: Onde fica toda a lógica para a solução do problema (Classes POJO)

- Persistência: Responsável por toda a lógica de persistência dos dados (DAO's).

- Dados: A qual é representada pelo banco de dados.

Resumindo, qualquer implementação futura que utilizar este arcabouço, terá tanto os benefícios do padrão MVC, quanto da separação em camadas, pois os mesmos são utilizados neste arcabouço.

3. Acoplamento, clareza e organização do código fonte

O Acoplamento é dado pelo grau de dependência entre as classes no código fonte. Quanto maior for este, mais difícil será dar manutenção a este código.

No que diz respeito a estes três aspectos, o arcabouço apresenta boas clareza e organização de uma forma geral. Pecando um pouco na camada de apresentação, especificamente, no que diz respeito ao modo como as folhas de estilo incidem nas páginas JSP, pois a parte de estilo das páginas está um pouco fragmentada, onde temos vários CSS's, nos quais muitos estilos aplicam-se a um mesmo objeto da página, através do uso de Id's, tornando um pouco difícil a administração e o reuso desta parte do código.

Com relação ao acoplamento, o código está bem dividido e com um baixo acoplamento.

4. Modelo de dados

O modelo de dados deste arcabouço é bastante simples e intuitivo, o que facilita tanto o entendimento quanto a administração.

5. Grau de conhecimento nas tecnologias utilizadas e dificuldades enfrentadas

Como já possuo alguma experiência em grande parte das tecnologias que foram utilizadas para a construção do arcabouço (Java, JSP, Struts, HTML, CSS, JBOSS, JBOSS AOP, Prototype, SQL e MySQL) e que o arcabouço está bem intuitivo, não tive grandes dificuldades em implementar as funcionalidades pedidas. Porém, a parte de folhas de estilo e páginas JSP está um pouco confusa, conforme já disse acima.

6. Tecnologias utilizadas para a construção do arcabouço

Dado que todas as tecnologias utilizadas na construção deste arcabouço são gratuitas, e em sua maioria são consolidadas, futuras implementações de instâncias de software que apóiem *group storytelling*, serão facilitadas.

7. Considerações finais

Apesar deste arcabouço, ser um projeto pioneiro na área de *group storytelling*, necessitando ser melhor testado, para que sujam pontos de aperfeiçoamento, acredito que seja um bom ponto de partida para a construção de instâncias de software que apóiem a técnica *group storytelling*, pois sua utilização pode agilizar o processo de construção de tais instâncias. Seria interessante também acrescentar um apêndice explicando como configurar o projeto no Eclipse.

7.3.2 Considerações sobre o relatório de *feedback*

Um relatório de *feedback* na forma de texto livre é uma maneira bem flexível de se ter uma avaliação sob outro olhar. O ideal seria que essa avaliação fosse feita sob diversos pontos de vista, por pessoas com diferentes perfis. Entretanto, fizemos esta avaliação com apenas uma pessoa dada a escassez de tempo para conclusão do trabalho.

A pessoa que fez essa avaliação e que, portanto, redigiu esse relatório, é um membro do grupo de pesquisa Greco. O avaliador tem um perfil de desenvolvedor de software. Se

tivéssemos optado por avaliar o arcabouço com a ajuda de uma pessoa com outro perfil, obteríamos outra visão de uso do arcabouço, mas isso demandaria mais tempo.

O relatório apresenta as observações do avaliador citando tanto pontos positivos como aspectos negativos segundo a visão do mesmo. Fica claro no texto do relatório que o avaliador observou com mais detalhes os aspectos técnicos do arcabouço. Essas observações certamente devem ser consideradas em trabalhos futuros, onde novas versões do arcabouço serão desenvolvidas, para corrigir e aperfeiçoar a versão atual.

7.4 Considerações gerais

Analisando os resultados das avaliações executadas, podemos identificar indícios de que o arcabouço tem potencial para reduzir o trabalho de desenvolvimento de ferramentas de *group storytelling*. Na avaliação baseada em métricas esses indícios ficam expressos nos números que foram apresentados na Tabela 11. A manutenção de alguns valores ou os aumentos na faixa entre 15% e 18% significam que boa parte do código foi reutilizada.

Na avaliação com relatório de *feedback* os indícios se apresentam em frases tais como: “o mesmo pode ser reutilizado, já que é genérico, agilizando o processo de construção dessa instância”, “qualquer implementação futura que utilizar este arcabouço, terá tanto os benefícios do padrão MVC, quanto da separação em camadas” ou “Dado que todas as tecnologias utilizadas na construção deste arcabouço são gratuitas, e em sua maioria são consolidadas, futuras implementações de instâncias de software que apoiem *group storytelling*, serão facilitadas”.

Apesar dessas avaliações já apresentarem os indícios de redução do esforço no desenvolvimento de novas ferramentas, é importante que outras avaliações sejam feitas. A instanciação do arcabouço por outros avaliadores com diferentes perfis traria outros pontos de vista. A utilização de uma instância do arcabouço numa dinâmica de *group storytelling* seria

outro tipo de avaliação que poderia expor alguns aspectos não observados nas nossas avaliações.

8 Conclusão

Neste capítulo faremos uma breve revisão do que foi feito neste trabalho, enumerando contribuições alcançadas e limitações encontradas. No final serão apresentadas questões que consideramos importantes e que poderão ser tratadas em trabalhos futuros.

8.1 Considerações finais

Neste trabalho propusemos um arcabouço para ser usado no desenvolvimento de ferramentas de apoio ao *group storytelling*. Para elaborarmos essa proposta iniciamos a pesquisa fazendo um estudo do domínio de gestão do conhecimento, analisando aspectos ligados a esta área, principalmente a questões relacionadas ao trabalho cooperativo e à recuperação coletiva do conhecimento. Em seguida, fizemos um levantamento de soluções de *group storytelling* já implementadas assim como de outras soluções relacionadas ao mesmo tema. Os requisitos foram observados sob o ponto de vista de *groupware*, com foco nos aspectos de cooperação, coordenação e comunicação (modelo 3C). Neste momento começamos a identificar o problema das constantes re-implementações, o que deu foco ao próximo passo de pesquisa.

Após a identificação do problema, decidimos elaborar um arcabouço como proposta de solução, e iniciamos fazendo um estudo sobre arcabouços, principalmente em metodologias de desenvolvimento destes artefatos. Foi importante também o estudo sobre classificações de arcabouço o que nos ajudou a situarmos aonde se encaixava a nossa proposta.

Toda a análise do domínio feita no primeiro passo, associada ao estudo sobre arcabouços, foram essenciais para elaborarmos o Sofia como proposta desta dissertação. Ele agrega funcionalidades provenientes dos trabalhos estudados, características de sistemas de *groupware* e conceitos clássicos de engenharia de software.

Dentre as contribuições deste trabalho podemos citar:

- O levantamento, a análise e a comparação entre as diferentes abordagens de uso do *group storytelling* como técnica de gestão do conhecimento em diferentes contextos;
- O projeto conceitual de um arcabouço para construção de ferramentas de *group storytelling* baseado em cinco principais alicerces:
 - *Groupware* baseado no modelo 3C
 - Desenvolvimento baseado em componentes
 - Padrões de projeto
 - Mashup
 - Código aberto
- O desenvolvimento do software que torna concreto o projeto conceitual e que permitiu a avaliação baseada em instâncias;
- A inclusão de funcionalidades de coordenação dos contadores e de associação de outras mídias às histórias. Os trabalhos de *group storytelling* estudados mencionavam a importância dessas funcionalidades, mas suas soluções finais as consideravam como trabalhos futuros.

O arcabouço proposto neste trabalho pode ser usado tanto na sua forma conceitual, aproveitando-se da modelagem que apresentamos, como na sua forma concreta, com o uso do software que foi desenvolvido. No primeiro caso seria necessário todo o desenvolvimento do código obedecendo às características da modelagem que foi proposta. O segundo caso é o mais comum e existem duas maneiras de se utilizar o Sofia para aplicá-lo em situações onde se quer fazer uso do *group storytelling* para gestão do conhecimento.

A primeira delas não exige que se manipule o código do sistema e dessa maneira a pessoa está limitada a utilizar as opções de configuração e a flexibilidade oferecida pelo arcabouço na sua versão base. Isso será feito principalmente por aquelas pessoas que não são da área de computação ou que não necessitam de funcionalidades além das que já estão disponíveis. Nesse caso, o arcabouço permite que sejam feitas configurações que afetam todas as histórias do sistema ou apenas histórias específicas.

A segunda maneira consiste na instanciação do arcabouço com a inclusão de novas funcionalidades o que implica na manipulação do código. Nesse caso é necessário que se compreenda a arquitetura do arcabouço e que se tenha conhecimento sobre as tecnologias utilizadas no desenvolvimento do mesmo. Isso exige um esforço maior por parte da pessoa que está fazendo a instanciação, mas ela consegue uma flexibilidade maior de modificação e inclusão de novas características.

Reconhecemos que o arcabouço proposto é apenas um primeiro passo no sentido de evitar o re-trabalho no desenvolvimento de soluções de apoio ao GST. Outros esforços de modificação e melhoria terão que ser executados para que o arcabouço ganhe maturidade. Questões que consideramos importantes e que poderão ser tratadas em trabalhos futuros são:

- Agregar um mecanismo de busca que facilite o acesso dos usuários a todo tipo de conteúdo que é gerado durante as dinâmicas. Isso inclui uma avaliação do que deve ser indexado e como esses conteúdos deverão ser procurados.
- Melhorar a validação do código incluído pelo usuário no sistema na parte de mashup de conteúdos. Essa é uma melhoria que envolve pesquisa relacionada à segurança de sistemas;
- Agregar novas funcionalidades de coordenação, além das já implementadas, que ajudem o facilitador de uma história a conduzir a dinâmica;

- Agregar novas funcionalidades de comunicação que possam ser usadas nas dinâmicas que exigem esses recursos extras. O arcabouço já prevê essa flexibilidade, mas ele não inclui essas funcionalidades, pois alguns autores consideram que o uso de outros meio de comunicação durante a dinâmica de GST promove fuga de conhecimento que deveria estar contido na história.
- Avaliação mais ampla do arcabouço, tanto com abordagens objetivas e formais, como com abordagens subjetivas com diferentes perfis de usuários do arcabouço. Essa avaliação indicaria pontos falhos e novos requisitos não contemplados na versão atual.

A natureza modular e baseada em componentes do arcabouço permite que ele seja estendido através da inclusão de novas funcionalidades que complementam a nossa proposta. Todas essas melhorias podem ser implementadas por quem está instanciando o arcabouço, mas elas podem facilmente ser agregadas à versão base do arcabouço para que novas instanciações não façam esse esforço novamente.

Referências

ALMEIDA, E. S. et al. **CRUISE** – component reuse in software engineering. 2007 Disponível em: <http://cruise.cesar.org.br/>. Acesso em outubro 2007.

ALUR, D. ; CRUPI, J. ; MALKS, D. **Core J2EE patterns** – best practices and design strategies. Englewood Cliffs: Prentice Hall, 2001. ISBN 0-1306-4884-1.

BORGES, M. R. S. ; PINO, J. A. Awareness mechanisms for coordination in asynchronous CSCW. In: WORKSHOP ON INFORMATION TECHNOLOGIES AND SYSTEMS, 9., 1999, Charlotte. **Proceedings ...** Charlotte: University of North Caroline, Charlotte, 1999. p. 69-74.

BURBECK, S. **Applications programming in smalltalk-80(TM)**: how to use model-view-controller (MVC). 1992. Disponível em: <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>. Acesso em: nov. 2007.

BUSCHMANN, F. et al. **Pattern-oriented software architecture**: a system of patterns. New York: John Wiley and Sons, 1996. ISBN 0-4719-5869-7.

CARMINATTI, N. ; BORGES, M. R. S. ; GOMES, J. O. Analysing approaches to collective knowledge recall. **Computing and Informatics**, Bratislava, v. 25, n. 6, p. 547-570, 2006.

CARMINATTI, N. **Recuperação coletiva de conhecimento aplicada a situações de emergência**. 2006. Dissertação (Mestrado em Informática) – Instituto de Matemática / Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2006.

CHRISTOPHER, A. et al. **A pattern language**. New York: Oxford University Press, 1997. ISBN: 0-1950-1919-9.

DANDELIFE. **Dandelife - your lifecastTM**: your photos, videos and blog mixed with your friends, family and feeds. Disponível em: <http://dandelife.com/> Acesso em: maio de 2007.

DAVENPORT, T. H. ; PRUSAK, L. **Working knowledge**: how organizations manage what they know. Boston: Harvard Business Press, 1998.

DAVID, J. M. N. **Um serviço de percepção para uma infra-estrutura de desenvolvimento de groupware**. 2004. Tese (Doutorado em Engenharia de Sistemas e Computação) Coordenação do Programa de Pós-Graduação de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

DBDESIGNER. **DBDesigner**. Disponível em: <http://www.fabforce.net/dbdesigner4/> Acesso em: maio 2008.

EARLES, J. **Framework evolution!** one box, two box, white box, black box. Disponível em: http://www.cbd-hq.com/articles/2000/000401je_frameworks2.asp. Acesso em: out. 2007.

ECLIPSE. **Eclipse** - an open development platform. 2007. Disponível em: <http://www.eclipse.org/>. Acesso em: out. 2007.

EJB. **Enterprise javabeans technology**. Disponível em: <http://java.sun.com/products/ejb/>. Acesso em: nov. 2007.

ELLIS, C. A. ; GIBBS, S. J. ; REIN, G. L. Groupware: some issues and experiences. **Communications of the ACM**, New York, v. 34, n. 1, p. 39-58, Jan. 1991.

ERTL, A. **How popular are various programming languages?** 2007. Disponível em: <http://www.complang.tuwien.ac.at/anton/comp.lang-statistics/>. Acesso em: nov. 2007.

FAYAD, M. E. ; SCHMIDT, D. C. Object-oriented application frameworks. **Communications of the ACM**, New York, v.. 40, n. 10, p. 32-38, Oct., 1997.

FOWLER, M. **Analysis patterns: reusable object models**. Reading: Addison-Wesley, 1997. ISBN 0-2018-9542-0.

FUKS, H. et al. Applying the 3C model to groupware development. **International Journal of Cooperative Information Systems**, London, v.14, n. 2-3, p. 299-328, Jun-Sept 2005, ISSN 0218-8430.

GAMMA, E. et al. **Design patterns: elements of reusable object-oriented software**. Reading: Addison-Wesley, 1995. ISBN 0-2016-3361-2.

GBDIRECT. **Benefits of using open source software**. 2007 Disponível em: <http://open-source.gbdirect.co.uk/migration/benefit.html>. Acesso em: out. 2007.

GRECO. **Grupo de Engenharia do Conhecimento**. Disponível em: <http://www.greco.nce.ufrj.br>. Acesso em: out. 2008.

HENDERSON-SELLERS, B. **Object-oriented metrics: measures of complexity**. Englewood Cliffs: Prentice-Hall, 1996. ISBN 0-1323-9872-9.

JAVA. **Sun microsystems.** the Java 2 platform. 1995. Disponível em: <http://java.sun.com/j2se>. Acesso em: out. 2007.

JAVA. **The Java language.** an overview. 2007. Disponível em: <http://java.sun.com/docs/overviews/java/java-overview-1.html>. Acesso em: nov. 2007.

JBOSS. **JBoss application aerver.** 2007. Disponível em: <http://www.jboss.org/products/jbossas>. Acesso em: out. 2007.

JOHNSON, R. E. ; FOOTE, B. Designing reusable classes. **Journal of Object-Oriented Programming**, New York, v.1, n. 2, p. 22-35. jun./ jul. 1988.

JOHNSON, R. E. **How to design frameworks.** 1993. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/recursos/how-design-frame.pdf>. Acesso em: out. 2007.

KRUEGER, C. W. Software reuse. **ACM Computing Surveys**, New York, v. 24, n. 2, p. 131-183, Jun. 1996.

LABELLE, F. **Programming language usage graph.** 2006. Disponível em: <http://www.cs.berkeley.edu/~flab/languages.html>. Acesso em: nov. 2007.

LAPORTI, V. ; BORGES, M. ; BRAGANHOLO, V. A Collaborative approach to requirements elicitation. 2007. Melbourne. **Proceedings ...** Melbourne: Swinburne University of Technology / IEEE, 2007.

LARMAN, C. **Applying UML and patterns:** an introduction to object-oriented analysis and design and the unified process. Englewood Cliffs: Prentice Hall PTR, 2002. ISBN 0-1309-2569-1.

LUZ, C. M. ; BORGES, M. R. S., CAMPOS, M. L. M. **Sofia – a framework for the development of Group Storytelling Tools.** ACM WORKSHOP ON STORY REPRESENTATION, MECHANISM AND CONTEXT, 2., 2008, Vancouver. **Proceedings ...** Vancouver, 2008. Aceito para publicação.

MARKIEWICZ, M. E. ; LUCENA, C. J. P. **Object oriented framework development.** Disponível em: <http://www.acm.org/crossroads/xrds7-4/frameworks.html>. Acesso em: out. 2007.

MARTIN, R. Object oriented design quality metrics: an analysis of dependencies, **ROAD**, v. 2, n., 3, Sep.-Oct, 1995.

MASHUP. Mashup (web application hybrid). 2007. **Wikipedia**. Disponível em: [http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)). Acesso em: out. 2007.

METRICS. **Metrics**, 2008. Disponível em: <http://metrics.sourceforge.net/>. Acesso em: jul. 2008.

MYSQL. **MySQL - the world's most popular open source database**, 2008 Disponível em: <http://www.mysql.com/>, Acesso em: abr. 2008.

NONAKA, I. The knowledge-creating company. In: DRUCKER, P. F. **Harvard Business Review on knowledge management**. Boston: Harvard Business Scholl Press, 1998. p. 21-45, (Harvard Business Review Paperback Series). ISBN 0-87584-881-8.

NONAKA, I. ; TAKEUCHI, H. **Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação**. 2 ed. Rio de Janeiro: Campos, 1997. ISBN 8-5352-0177-7.

OLIVEIRA, A. **Histórias coletivas de uso de sistemas: capturando conhecimento tácito para manutenção**. 2006. Dissertação (Mestrado em Informática) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

O'REILLY, T. **Programming language wars**. Part one. 2007. Disponível em: http://radar.oreilly.com/archives/2007/03/programming_lan.html. Acesso em: nov. 2007.

OURSTORY. **OurStory.com - save stories, photos, and videos on a collaborative timeline**. Disponível em: <http://www.ourstory.com/>. Acesso em: maio 2007.

PATTERNS. **J2EE patterns catalog**. Disponível em: <http://java.sun.com/blueprints/patterns/catalog.html>. Acesso em: nov. 2007.

PERRET, R. **A técnica de group storytelling aplicada à gestão do conhecimento**. Dissertação 2004. (Mestrado em Informática) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

POLANYI, M. **The tacit dimension**. London: Routledge & Kegan Paul, 1966.

PREE, W. **Design patterns for object oriented software development**. Reading: Addison-Wesley, 1995. ISBN 0-2014-2294-8.

PROTOTYPE. **Prototype - javascript framework** Disponível em: <http://www.prototypejs.org/> Acesso em: maio 2008.

REIS, M. ; BORGES, M. R. S. ; GOMES, J. Recalling resilient actions during emergency response. In: LÖFFLER, J. ; KLANN, M (Ed.). **Mobile Response: International Workshop on Mobile Information Technology for Emergency Response**, Sankt Augustin. 2007. Revised Selected Papers. Berlin: Springer, 2008.

SCHREIBER, G. et al. **Knowledge engineering and management – the commonKADS methodology**. Londres: MIT Press, 1999. ISBN 0-262-19300-0.

SHARMA, N. **The origin of the “data information knowledge wisdom” hierarchy**. Disponível em: http://www-personal.si.umich.edu/~nsharma/dikw_origin.htm Acesso em: jun. 2008

SILVA, R. P. **Suporte ao desenvolvimento e uso de frameworks e componentes**. 2000. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

SILVA, S. A. **Modelação de processos alternativos de conflito utilizando a técnica de storytelling**. 2006. Dissertação (Mestrado em Engenharia Informática e de Computadores) – Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, 2006.

STORYSTREAM. **StoryStream** – make stories. Disponível em: <http://www.iwonderdesigns.com/storystream/>. Acesso em: jun. 2008.

STRUTS. **Apache struts** – web application framework. Disponível em: <http://struts.apache.org/>. Acesso em: nov. 2007.

SUTHERLAND, J. **The separation of structure, presentation, and behaviour as a software architecture**. Disponível em: <http://www.mercurytide.co.uk/whitepapers/separating-structure-presentation-and-behaviour/>. Acesso em: maio 2008.

TALIGENT. **Building object-oriented frameworks**. Taligent, 1994. Disponível em: <http://lhcb-comp.web.cern.ch/lhcb-comp/Components/postscript/buildingoo.pdf>. Acesso em: out. 2007.

TERRA, J. C. C. ; GORDON, C. **Portais corporativos: a revolução na gestão de conhecimento**. São Paulo: Negócio, 2002.

TILES. **Tiles**. Disponível em: <http://www2.lifl.fr/~dumoulin/tiles/>. Acesso em: nov. 2007.

TRAVASSOS, G. H. ; GUROV, D. ; AMARAL, E. A. G. **Introdução a engenharia de software experimental**. Rio de Janeiro: COPPE/UFRJ, 2002. (Relatório Técnico PESC 590/02).

VALLE, C. ; PRINZ, W. ; BORGES, M. R. S. Generation of group storytelling in post-decision implementation process. In: INTERNATIONAL CONFERENCE ON CSCWD, 7., 2002. Rio de Janeiro. **Proceedings ...** Rio de Janeiro: SBC, 2002. p. 361-367.

VIADIGITAL. **Tecnologias requeridas para construção de componentes via digital/FLO-PREF**. Disponível em: http://repositorio.viadigital.org.br/frs/download.php/83/Anexo_A4_tecnologia.doc. Acesso em: nov. 2007.

WEB2.0. Web 2.0. 2004. **Wikipedia**. Disponível em: http://en.wikipedia.org/wiki/Web_2.0. Acesso em: out. 2007.

WELTON, D. N. **Programming language popularity**. 2005. Disponível em: http://www.welton.it/articles/language_popularity.html. Acesso em: nov. 2007.

YACOUB, S. ; AMMAR, H. H. Towards pattern oriented frameworks. **Journal of Object Oriented Programming**, Brentwood: v. 12, n. 8, p.25-35, Jan. 2000.

ZELNY, M. **Human systems management** – integrating knowledge, management and systems. London: World Scientific Publishing, 2005.

ZELNY, M. Knowledge-information autopoietic cycle: towards the wisdom systems. **International Journal of Management and Decision Making**, Geneve, v.. 7, n. 1, p.3–18. 2006

Anexo 1 – Preparação do ambiente

Este anexo visa ser um guia para preparação do ambiente onde o arcabouço será instalado. Apresentaremos os passos de preparação da base de dados, de instalação e configuração do servidor de aplicações, e de implantação do arcabouço neste ambiente. Os sistemas operacionais considerados são o Windows e o Ubuntu.

A1 Instalação dos sistemas de gerência da base de dados

A1.1 Instalação do MySQL

A versão do MySQL usada no desenvolvimento e nos testes do arcabouço foi a 5.0. Não foram feitos testes com outras versões.

Windows

A instalação no Windows é trivial bastando que seja obtido no site o pacote chamado de `essential`, que inclui um passo-a-passo de instalação.

Ubuntu

Execute dentro de um terminal o comando:

```
sudo apt-get install mysql-server-5.0
```

O processo de instalação pedirá para que se defina a senha de root do banco antes de terminar.

A1.2 Instalação das ferramentas gráficas para manipulação do banco

Nesse passo serão instaladas as ferramentas gráficas MySQL Administrator e MySQL Query Browser.

Windows

Obter do site o pacote `mysql-gui-tools` e fazer a instalação seguindo o passo-a-passo.

Ubuntu

Execute dentro de um terminal os comandos:

```
sudo apt-get install mysql-admin
```

```
sudo apt-get install mysql-query-browser
```

Após a instalação destes dois pacotes, o Ubuntu disponibilizará no menu de aplicações as opções para execução dos mesmos.

A2 Criação e preparação da base de dados

A2.1 Criar o Schema

Entrar no MySQL Administrator, escolher o item Catalogs e criar um *schema* com nome `sofia`.

Entrar em seguida no item User Administration e criar um novo usuário de nome `sofia` com senha `sofia`. Clicar sobre o usuário e adicionar o host `localhost` para o mesmo. Excluir o host `%` (que significa qualquer host) para esse usuário (essa alteração faz com que apenas conexões vindas de `localhost` sejam permitidas. Além disso, resolve um *bug* que aconteceu nos testes). Associar os privilégios `SELECT`, `INSERT`, `UPDATE` E `DELETE` sobre o schema `sofia` para esse usuário.

Entrar no MySQL Query Browser usando o usuário `root` escolhendo o Default Schema `sofia`. Executar o script `create_sofia.sql` para criar as tabelas e outros artefatos e em seguida, o script `insert_sofia.sql`, para fazer a carga inicial no banco.

A3 Instalação do servidor de aplicações

A3.1 Instalação do JBoss

A versão de JBoss utilizada no desenvolvimento e nos testes do arcabouço foi a 4.2.2. A instalação do JBoss é trivial bastando obter o arquivo do site e descompactá-lo na máquina. O JBoss possui 3 modos de execução: minimal, default e all. O arcabouço exige que seja utilizado o modo all, pois apenas nesse modo as funcionalidades ligadas a AOP estão habilitadas. No texto usaremos <jboss_home> para fazer referência ao local de instalação do JBoss.

A4 Configuração dos serviços no servidor de aplicações

A4.1 Configurando o datasource

Para configurar o datasource que dará acesso ao banco de dados sofia basta que o arquivo `sofia-mysql-ds.xml` seja copiado para o diretório de deploy do JBoss <jboss_home>\server\all\deploy\. Também é necessário que seja copiado para o diretório <jboss_home>\server\all\lib\ o arquivo jar com as classes do driver de acesso ao banco. No MySql 5 esse arquivo se chama `mysql-connector-java-5.0.4-bin.jar`.

A4.2 Configurando o serviço de e-mail

O serviço de email que o arcabouço utiliza é configurado também através da cópia de um arquivo xml que define os detalhes de conexão. Para disponibilizar o serviço é necessário

que seja feita a cópia do arquivo `sofia-mail-service.xml` para o diretório de deploy do JBoss (`<jboss_home>\server\all\deploy\`).

A4.3 Habilitando AOP no JBoss

São 3 os requisitos para que o AOP utilizado pelo arcabouço funcione:

1 – Habilitar o LoadtimeWeaving

Alterar o arquivo `<jboss_home>\server\all\deploy\jboss-aop-jdk50.deployer\META-INF\jboss-service.xml` mudando a linha

```
<attribute name="EnableLoadtimeWeaving">false</attribute>
```

pela linha

```
<attribute name="EnableLoadtimeWeaving">true</attribute>
```

2 – Disponibilizar para o JBoss o pluggable-instrumentor

Copiar o arquivo `pluggable-instrumentor.jar` para pasta `<jboss_home>\bin\`

Se estiver executando o JBoss por `run.sh`, alterar o arquivo `run.conf` adicionando a linha

```
JAVA_OPTS=" $JAVA_OPTS -javaagent:pluggable-instrumentor.jar"
```

Se estiver executando o JBoss por `run.bat`, adicionar as seguintes linhas ao começo do arquivo `run.bat`:

```
set JBOSS_CLASSPATH=%DIRNAME%pluggable-instrumentor.jar
set JAVA_OPTS=-javaagent:pluggable-instrumentor.jar
```

3 – Rodar o JBoss no modo all

Ao iniciar o JBoss, garantir que ele será executado no modo all.

```
run.sh -c all
```

ou

```
run.bat -c all
```

Obs.: Esse passo-a-passo de configuração do AOP encontram-se mais bem detalhado no link: <http://wiki.jboss.org/wiki/ConfiguringAopLoadtimeWeaving>

A5 Implantação e teste do arcabouço

Após a preparação do ambiente, para que seja feita a implantação do arcabouço, basta que seja copiado o arquivo Sofia.war para o diretório de deploy do servidor (<jboss_home>\server\all\deploy\). O teste da instalação é feito com o acesso ao sistema pela url <http://localhost:8080/Sofia> e com o login sendo feito com o email utilizado no script de carga inicial da aplicação com senha admin.

Script de criação das tabelas (create_sofia.sql)

```
CREATE TABLE acesso (  
  historia_id INTEGER UNSIGNED NOT NULL,  
  usuario_id INTEGER UNSIGNED NOT NULL,  
  perfil_relativo_id INTEGER UNSIGNED NOT NULL,  
  estado CHAR NOT NULL,  
  INDEX permissao_FKIndex2(usuario_id),  
  INDEX acesso_FKIndex2(historia_id),  
  INDEX acesso_FKIndex3(perfil_relativo_id)  
);
```

```
CREATE TABLE assoc_frag_xref (  
  tipo_assoc_id INTEGER UNSIGNED NOT NULL,  
  fragmento1_id INTEGER UNSIGNED NOT NULL,  
  fragmento2_id INTEGER UNSIGNED NOT NULL,
```

```
descricao TEXT NOT NULL,  
INDEX assoc_frag_xref_FKIndex1(fragmento1_id),  
INDEX assoc_frag_xref_FKIndex2(fragmento2_id),  
INDEX assoc_frag_xref_FKIndex3(tipo_assoc_id)  
);  
  
CREATE TABLE aviso (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
historia_id INTEGER UNSIGNED NOT NULL,  
titulo VARCHAR(255) NOT NULL,  
conteudo TEXT NOT NULL,  
data_cadastro DATETIME NOT NULL,  
PRIMARY KEY(id),  
INDEX aviso_FKIndex1(historia_id)  
);  
  
CREATE TABLE categoria (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
historia_id INTEGER UNSIGNED NOT NULL,  
tipo_frag_id INTEGER UNSIGNED NOT NULL,  
nome VARCHAR(45) NOT NULL,  
PRIMARY KEY(id),  
INDEX categoria_FKIndex1(tipo_frag_id),  
INDEX categoria_FKIndex2(historia_id)  
);  
  
CREATE TABLE categoria_default (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
tipo_frag_default_id INTEGER UNSIGNED NOT NULL,  
nome VARCHAR(45) NOT NULL,  
PRIMARY KEY(id),  
INDEX categoria_FKIndex1(tipo_frag_default_id)  
);  
  
CREATE TABLE comentario (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
usuario_id INTEGER UNSIGNED NOT NULL,  
fragmento_id INTEGER UNSIGNED NOT NULL,  
conteudo TEXT NOT NULL,  
data_cadastro DATETIME NOT NULL,  
PRIMARY KEY(id),  
INDEX comentario_FKIndex1(fragmento_id),  
INDEX comentario_FKIndex2(usuario_id)  
);  
  
CREATE TABLE fragmento (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
usuario_id INTEGER UNSIGNED NOT NULL,  
historia_id INTEGER UNSIGNED NOT NULL,  
tipo_frag_id INTEGER UNSIGNED NOT NULL,
```

```
categoria_id INTEGER UNSIGNED NULL,  
titulo VARCHAR(45) NOT NULL,  
conteudo TEXT NOT NULL,  
data_cadastro DATETIME NOT NULL,  
anonimo CHAR NOT NULL,  
estado CHAR NULL,  
PRIMARY KEY(id),  
INDEX fragmento_FKIndex1(usuario_id),  
INDEX fragmento_FKIndex2(historia_id),  
INDEX fragmento_FKIndex3(tipo_frag_id),  
INDEX fragmento_FKIndex4(categoria_id)  
);  
  
CREATE TABLE glossario (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
historia_id INTEGER UNSIGNED NOT NULL,  
termo VARCHAR(45) NOT NULL,  
descricao VARCHAR(255) NOT NULL,  
PRIMARY KEY(id, historia_id),  
INDEX glossario_FKIndex1(historia_id)  
);  
  
CREATE TABLE historia (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
usuario_id INTEGER UNSIGNED NOT NULL,  
titulo VARCHAR(45) NOT NULL,  
descricao TEXT NOT NULL,  
estado CHAR NOT NULL,  
data_cadastro DATETIME NOT NULL,  
data_finalizacao DATETIME NULL,  
tipo CHAR NOT NULL,  
anonimato CHAR NOT NULL,  
anexos CHAR NOT NULL,  
comunicacao_extra CHAR NOT NULL,  
auto_gerencivel CHAR NOT NULL,  
PRIMARY KEY(id),  
INDEX estoria_FKIndex1(usuario_id)  
);  
  
CREATE TABLE midia (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
historia_id INTEGER UNSIGNED NOT NULL,  
usuario_id INTEGER UNSIGNED NOT NULL,  
tipo_midia_id INTEGER UNSIGNED NOT NULL,  
titulo VARCHAR(45) NOT NULL,  
descricao VARCHAR(255) NOT NULL,  
conteudo TEXT NOT NULL,  
data_cadastro DATETIME NOT NULL,  
PRIMARY KEY(id),  
INDEX midia_FKIndex1(historia_id),
```

```

INDEX midia_FKIndex2(tipo_midia_id),
INDEX midia_FKIndex3(usuario_id)
);

```

```

-----
-- Por enquanto usada apenas para definir se o usuário é administrador ou não
-----

```

```

CREATE TABLE perfil_absoluto (
  id INTEGER UNSIGNED NOT NULL,
  nome VARCHAR(45) NOT NULL,
  PRIMARY KEY(id)
);

```

```

-----
-- Tabelas de perfis que dependem da história com que o usuário está relacionado
-----

```

```

CREATE TABLE perfil_relativo (
  id INTEGER UNSIGNED NOT NULL,
  nome VARCHAR(45) NULL,
  PRIMARY KEY(id)
);

```

```

CREATE TABLE templates (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  nome VARCHAR(45) NOT NULL,
  descricao VARCHAR(255) NOT NULL,
  tipo CHAR NOT NULL,
  anonimato CHAR NOT NULL,
  anexos CHAR NOT NULL,
  comunicacao_extra CHAR NOT NULL,
  auto_gerenciavel CHAR NOT NULL,
  PRIMARY KEY(id)
);

```

```

CREATE TABLE tipo_assoc (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  historia_id INTEGER UNSIGNED NOT NULL,
  nome VARCHAR(45) NOT NULL,
  PRIMARY KEY(id),
  INDEX tipo_assoc_fragmento_FKIndex1(historia_id)
);

```

```

CREATE TABLE tipo_assoc_default (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  nome VARCHAR(45) NOT NULL,
  PRIMARY KEY(id)
);

```

```
CREATE TABLE tipo_frag (  
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  historia_id INTEGER UNSIGNED NOT NULL,  
  nome VARCHAR(45) NOT NULL,  
  PRIMARY KEY(id),  
  INDEX tipo_frag_FKIndex1(historia_id)  
);
```

```
CREATE TABLE tipo_frag_default (  
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  nome VARCHAR(45) NOT NULL,  
  PRIMARY KEY(id)  
);
```

```
CREATE TABLE tipo_midia (  
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  historia_id INTEGER UNSIGNED NOT NULL,  
  nome VARCHAR(45) NOT NULL,  
  PRIMARY KEY(id),  
  INDEX tipo_midia_FKIndex1(historia_id)  
);
```

```
CREATE TABLE tipo_midia_default (  
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  nome VARCHAR(45) NOT NULL,  
  PRIMARY KEY(id)  
);
```

```
CREATE TABLE usuario (  
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  email VARCHAR(255) NOT NULL,  
  senha VARCHAR(45) NOT NULL,  
  nome VARCHAR(45) NOT NULL,  
  data_cadastro DATETIME NOT NULL,  
  estado CHAR NOT NULL,  
  perfil_absoluto_id INTEGER UNSIGNED NOT NULL,  
  PRIMARY KEY(id),  
  INDEX usuario_FKIndex1(perfil_absoluto_id)  
);
```

```
CREATE TABLE usuario_acessou_comentario (  
  usuario_id INTEGER UNSIGNED NOT NULL,  
  comentario_id INTEGER UNSIGNED NOT NULL,  
  PRIMARY KEY(usuario_id, comentario_id),  
  INDEX usuario_has_comentario_FKIndex1(usuario_id),  
  INDEX usuario_has_comentario_FKIndex2(comentario_id)  
);
```

```
CREATE TABLE usuario_acessou_fragmento (  
  usuario_id INTEGER UNSIGNED NOT NULL,
```

```

    fragmento_id INTEGER UNSIGNED NOT NULL,
    PRIMARY KEY(usuario_id, fragmento_id),
    INDEX usuario_has_fragmento_FKIndex1(usuario_id),
    INDEX usuario_has_fragmento_FKIndex2(fragmento_id)
);

```

```

CREATE TABLE usuario_acessou_midia (
    usuario_id INTEGER UNSIGNED NOT NULL,
    midia_id INTEGER UNSIGNED NOT NULL,
    PRIMARY KEY(usuario_id, midia_id),
    INDEX usuario_has_midia_FKIndex1(usuario_id),
    INDEX usuario_has_midia_FKIndex2(midia_id)
);

```

```

-----
-- Se existe a relação aqui significa que o usuário já recebeu o alerta
-----

```

```

CREATE TABLE usuario_aviso_xref (
    usuario_id INTEGER UNSIGNED NOT NULL,
    aviso_id INTEGER UNSIGNED NOT NULL,
    PRIMARY KEY(usuario_id, aviso_id),
    INDEX usuario_has_aviso_FKIndex1(usuario_id),
    INDEX usuario_has_aviso_FKIndex2(aviso_id)
);

```

Script de carga inicial do banco (insert_sofia.sql)

```

INSERT INTO `perfil_absoluto` (`id`,`nome`) VALUES
(1,'administrador'),
(2,'usuario');

```

```

INSERT INTO `perfil_relativo` (`id`,`nome`) VALUES
(1,'Facilitador'),
(2,'Contador');

```

```

INSERT INTO `tipo_frag_default` (`id`,`nome`) VALUES
(1,'Relato'),
(2,'Fato'),
(3,'Descritor');

```

```

INSERT INTO `categoria_default` (`id`,`tipo_frag_default_id`,`nome`) VALUES
(1, 1, 'Opinião'),
(2, 1, 'Evento'),
(3, 1, 'Dado'),
(4, 1, 'Indefinida'),
(5, 1, 'Hipótese'),
(6, 3, 'Atividade'),
(7, 3, 'Indefinida'),

```

```
(8, 3, 'Local'),
(9, 3, 'Artefato'),
(10,3, 'Personagem');
```

```
INSERT INTO `tipo_midia_default` (`id`,`nome`) VALUES
(1,'Foto'),
(2,'Áudio'),
(3,'Video'),
(4,'Outro');
```

```
INSERT INTO `tipo_assoc_default` (`id`,`nome`) VALUES
(1,'Complementação'),
(2,'Temporal'),
(3,'Causa/Consequência'),
(4,'Negação/Contradição'),
(5,'Confirmação/Conformidade'),
(6,'Gap'),
(7,'Indefinida');
```

```
INSERT INTO `templates`
(`id`,`nome`,`descricao`,`tipo`,`anonimato`,`anexos`,`comunicacao_extra`,
`auto_gerenciablel`)
VALUES
(1, 'Privada identificada', '', 'R', 'N', 'U', 'N', 'N'),
(2, 'Privada com anonimato liberado', '', 'R', 'U', 'U', 'N', 'N'),
(3, 'Restrita', '', 'R', 'U', 'A', 'S', 'N'),
(4, 'Pública controlada', '', 'P', 'N', 'U', 'N', 'N'),
(5, 'Aberta ao público', '', 'P', 'N', 'M', 'S', 'N'),
(6, 'Pública auto gerenciada', '', 'P', 'N', 'M', 'S', 'S');
```

```
INSERT INTO `usuario` (`id`,`senha`,`nome`,`estado`,`data_cadastro`,`perfil_absoluto_id`,
`email`) VALUES
(1,'ISMvKXpXpadDiUoOSoAfww==','administrador','A', sysdate(), 1,
'sofia.pggi.nce@gmail.com');
```

Arquivo de configuração do datasource (sofia-mysql-ds.xml)

```
<?xml version="1.0" encoding="UTF-8"?>

<datasources>
  <local-tx-datasource>
    <jndi-name>MySQLDS</jndi-name>
    <connection-url>jdbc:mysql://localhost:3306/sofia</connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name>sofia</user-name>
    <password>sofia</password>
    <exception-sorter-class-
name>org.jboss.resource.adapter.jdbc.vendor.MySQLExceptionSorter</exception
-sorter-class-name>
    <metadata>
```

```

        <type-mapping>mySQL</type-mapping>
    </metadata>
</local-tx-datasource>
</datasources>

```

Arquivo de configuração do serviço de e-mail (sofia-mail-service.xml)

```

<?xml version="1.0" encoding="UTF-8"?>
<server>

    <!--
===== -->
    <!-- Mail Connection Factory
-->
    <!--
===== -->

    <mbean code="org.jboss.mail.MailService"
name="jboss:service=SofiaMail">
        <attribute name="JNDIName">java:/SofiaMail</attribute>
        <attribute name="User">sofia.ppgi.nce@gmail.com</attribute>
        <attribute name="Password">arcabouco</attribute>
        <attribute name="Configuration">
            <configuration>
                <property name="mail.store.protocol" value="pop3"/>
                <property name="mail.transport.protocol"
value="smtp"/>
                <property name="mail.user"
value="sofia.ppgi.nce@gmail.com"/>
                <property name="mail.pop3.host"
value="pop.gmail.com"/>
                <property name="mail.smtp.host"
value="smtp.gmail.com"/>
                <property name="mail.smtp.port" value="465"/>
                <property name="mail.smtp.user"
value="sofia.ppgi.nce@gmail.com"/>
                <property name="mail.smtp.password"
value="arcabouco"/>
                <property name="mail.smtp.auth" value="true"/>
                <property name="mail.smtp.starttls.enable"
value="true" />
                <property name="mail.from"
value="sofia.ppgi.nce@gmail.com"/>
                <property name="mail.debug" value="false"/>
                <property name="mail.smtp.socketFactory.class"
value="javax.net.ssl.SSLSocketFactory" />
                <property name="mail.smtp.socketFactory.port"
value="465" />
            </configuration>
        </attribute>
        <depends>jboss:service=Naming</depends>
    </mbean>
</server>

```

Anexo 2 – Modelo de dados

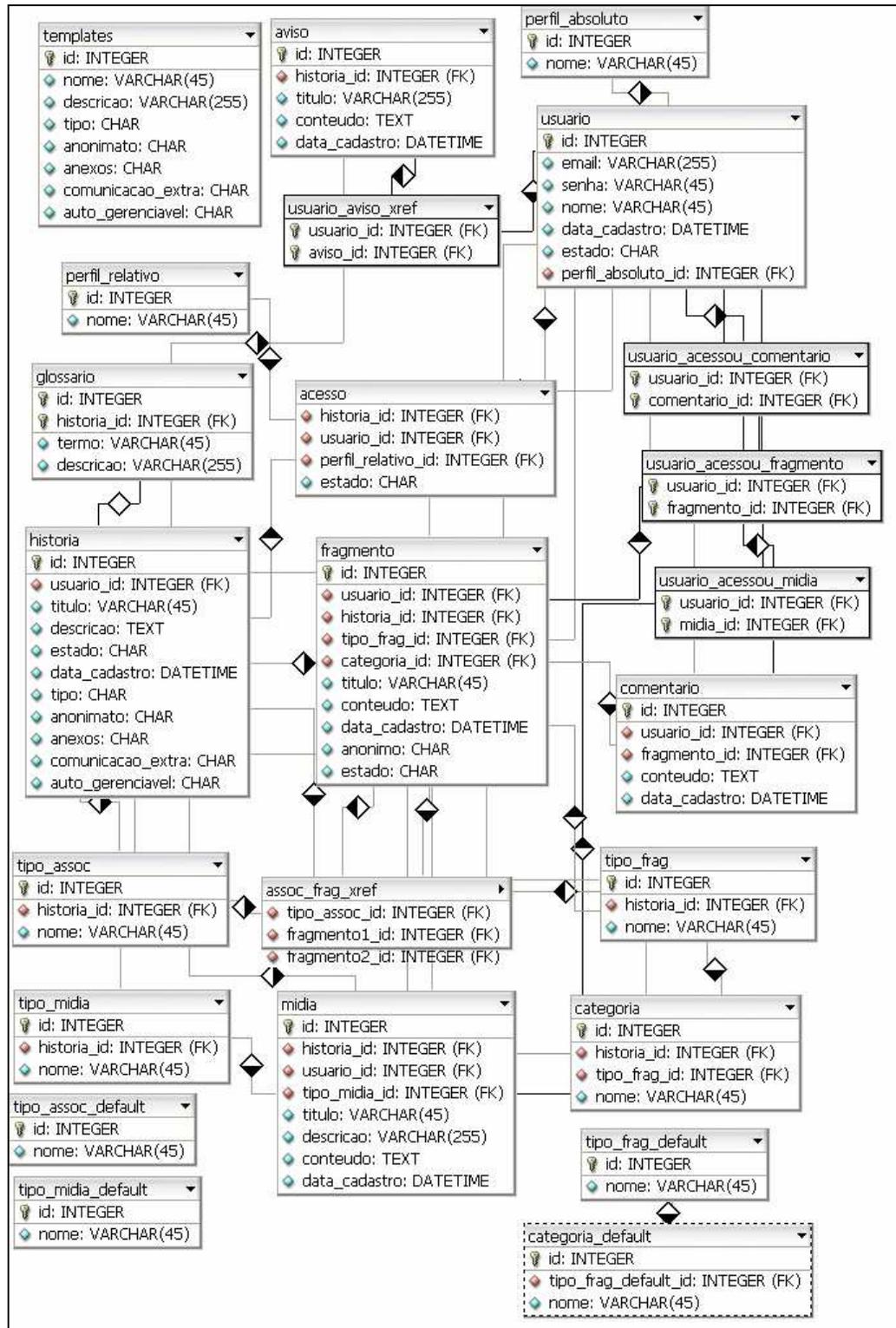


Figura 21 - Modelo de dados completo

Descrição de cada uma das entidades

acesso

Nessa tabela ficam registrados quais usuários tem acesso a cada uma das histórias e que tipo de perfil de acesso ele tem.

assoc frag xref

Nessa tabela ficam registradas as associações entre os fragmentos inclusive com a indicação do tipo de associação.

aviso

Nessa tabela ficam registrados os avisos cadastrados em cada uma das histórias.

categoria

Nessa tabela ficam registradas as categorias de fragmento que estão disponíveis para os contadores de uma história. Os contadores usam as categorias para classificar os fragmentos. As categorias estão associadas a um tipo de fragmento. O facilitador de uma história pode editar as categorias relacionadas a essa história existentes nessa tabela.

categoria default

Nessa tabela ficam registradas todas as categorias que são consideradas padrão para todas as histórias. Quando uma nova história é criada, os registros dessa tabela são copiados para tabela **categoria** ficando disponíveis para os contadores utilizarem.

comentario

Nessa tabela ficam registrados os comentários que estão associados aos fragmentos.

fragmento

Nessa tabela ficam registrados todos os fragmentos de construção das histórias. Através dela podemos identificar autor, tipo, categoria, dentre outras informações.

glossario

Nessa tabela ficam registrados os termos de glossário que estão associados a uma história.

historia

Nessa tabela ficam registradas as histórias que estão cadastradas no sistema. A tabela inclui, além de atributos básicos como título e data de cadastro, colunas que descrevem como a história está configurada tais como: tipo, anonimato, anexos, etc.

midia

Nessa tabela ficam registradas todas as mídias que foram associadas a uma história.

perfil absoluto

Nessa tabela ficam registrados os perfis absolutos de acesso, ou seja, o perfil de um usuário perante o sistema como um todo e não apenas perante uma história.

perfil relativo

Nessa tabela ficam registrados os perfis relativos de acesso, ou seja, os perfis que os usuários podem ter perante cada uma das histórias.

templates

Nessa tabela ficam registrados os modelos de configuração das histórias. Um usuário que está criando uma história pode selecionar um dos modelos que estão disponíveis nessa tabela ou então configurar a história a seu critério.

tipo assoc

Nessa tabela ficam registrados os tipos de associação entre fragmentos que estão disponíveis para os contadores de uma história. O facilitador de uma história pode editar os tipos relacionados a essa história existentes nessa tabela.

tipo assoc default

Nessa tabela ficam registrados os tipos padrão de associação entre fragmentos para todas as histórias. Quando uma nova história é criada, os registros dessa tabela são copiados para tabela **tipo_assoc** ficando disponíveis para os contadores utilizarem.

tipo frag

Nessa tabela ficam registrados os tipos de fragmento que estão disponíveis para os contadores de uma história. O facilitador de uma história pode editar os tipos relacionados a essa história existentes nessa tabela.

tipo frag default

Nessa tabela ficam registrados os tipos padrão de fragmento para todas as histórias. Quando uma nova história é criada, os registros dessa tabela são copiados para tabela **tipo_frag** ficando disponíveis para os contadores utilizarem.

tipo midia

Nessa tabela ficam registrados os tipos de mídia que estão disponíveis para os contadores de uma história. O facilitador de uma história pode editar os tipos relacionados a essa história existentes nessa tabela.

tipo_midia_default

Nessa tabela ficam registrados os tipos padrão de mídia para todas as histórias. Quando uma nova história é criada, os registros dessa tabela são copiados para tabela **tipo_midia** ficando disponíveis para os contadores utilizarem.

usuario

Nessa tabela ficam registrados todos os usuários que estão cadastrados no sistema.

usuario_acessou_comentario

Nessa tabela ficam registrados quais comentários um usuário acessou.

usuario_acessou_fragmento

Nessa tabela ficam registrados quais fragmentos um usuário acessou.

usuario_acessou_midia

Nessa tabela ficam registrados quais mídias um usuário acessou.

usuario_aviso_xref

Nessa tabela ficam registrados quais avisos um usuário já leu.