

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

THIAGO PEREIRA RIQUE

**WEBQDA: UMA FERRAMENTA WEB COLABORATIVA PARA APOIAR A
ANÁLISE QUALITATIVA DE DADOS**

João Pessoa
2011

THIAGO PEREIRA RIQUE

**WEBQDA: UMA FERRAMENTA WEB COLABORATIVA PARA APOIAR A
ANÁLISE QUALITATIVA DE DADOS**

Dissertação apresentada ao Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, como requisito parcial para obtenção do título de Mestre em Informática (Sistemas de Computação).

Orientador: Prof. Dr. Álvaro Francisco de Castro Medeiros

João Pessoa
2011

R594w Rique, Thiago Pereira.
WebQDA: uma ferramenta Web colaborativa para apoiar a
análise qualitativa de dados / Thiago Pereira Rique.- João
Pessoa, 2011.
86f. : il.
Orientador: Álvaro Francisco de Castro Medeiros
Dissertação (Mestrado) – UFPB/CCEN
1. Sistemas de Computação. 2. Aplicativos CAQDAS.
3. Sistemas colaborativos. 4. Análise Qualitativa de Dados.
5. Redes Sociais.

UFPB/BC

CDU: 004(043)

1

Ata da Sessão Pública de Defesa de Dissertação de Mestrado do THIAGO PEREIRA RIQUE, candidato ao Título de Mestre em Informática na Área de Sistemas de Computação, realizada em 29 de março de 2011.

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

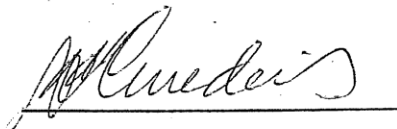
25

26

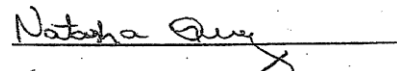
Aos vinte e nove dias do mês de março do ano dois mil e onze, às nove horas, na Sala de Reunião do Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, reuniram-se os membros da Banca Examinadora constituída para examinar o candidato ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa "Computação Distribuída", o Sr. Thiago Pereira Rique. A comissão examinadora composta pelos professores doutores: Álvaro Francisco de Castro Medeiros (DI - UFPB), Orientador e Presidente da Banca Examinadora, Natasha Correia Queiroz Lino (DI-UFPB) como examinador interno e Francisco Vilar Brasileiro (UFCG) como examinador externo. Dando início aos trabalhos, o Prof. Álvaro Francisco de Castro Medeiros, cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse, oralmente, a exposição do trabalho de dissertação intitulado "WEBQDA: UMA FERRAMENTA WEB COLABORATIVA PARA APOIAR A ANÁLISE QUALITATIVA DE DADOS". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "aprovado". Assim sendo, deve a Universidade Federal da Paraíba expedir o respectivo diploma de Mestre em Informática na forma da lei e, para constar, a professora Tatiana Aires Tavares, Sra. Coordenadora do PPGI, lavrou a presente ata, que vai assinada por ela, e pelos membros da Banca Examinadora. João Pessoa, 29 de março de 2011.


Tatiana Aires Tavares

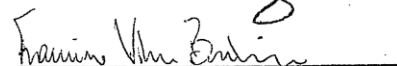
Prof. Dr. Álvaro Francisco de Castro Medeiros
Orientador (DI-UFPB)



Profª. Dra. Natasha Correia Queiroz Lino
Examinador Interno (DI-UFPB)



Prof. Dr. Francisco Vilar Brasileiro
Examinador Externo (UFCG)



27

Dedico este trabalho a todos aqueles que
contribuíram para sua realização.

AGRADECIMENTOS

A Deus, por tudo que tem feito na minha vida.

Gostaria de agradecer à minha família, pela constante torcida e suporte que me foi dado.

Ao meu orientador Álvaro Medeiros, pela amizade, competência e dedicação.

Aos grandes amigos Rodrigo Fujioka e Gustavo Sávio, por todo o auxílio que me foi dado durante a realização deste trabalho, além dos momentos de descontração que compartilhamos.

A Viviane Caires e Samara Martins, por sempre me incentivarem e pela amizade incondicional.

Aos amigos Pablo Queiroz e Janine Falcão, por fazerem a diferença em minha vida. Amo vocês, meus queridos.

A todos que de alguma forma estiveram presentes e me auxiliaram na elaboração deste trabalho.

RESUMO

É inegável a importância dos ambientes colaborativos no mundo globalizado, seja para possibilitar o compartilhamento de informações ou tornar possível a interação entre pessoas distantes. Também é fato, na sociedade atual, a necessidade de realização de tarefas e solução de problemas de forma colaborativa. Como exemplo, pode-se citar a pesquisa qualitativa que, quando realizada com o auxílio do computador, pode fazer uso dos aplicativos CAQDAS (*Computer Assisted Qualitative Data Analysis Software*). Apesar de ser possível realizar análises qualitativas em aplicativos CAQDAS de forma isolada, o estudo/trabalho realizado por um grupo possui o diferencial de permitir a interação entre os membros de uma equipe, possibilitando a expressão de pontos de vista e opiniões diferentes, além de ser mais propenso a comentários e críticas que contribuem para a melhoria e qualidade do trabalho. Desse modo, este documento apresenta o WebQDA, uma ferramenta colaborativa que utiliza as funcionalidades básicas da análise qualitativa de dados, visando ilustrar como os novos conceitos da Web 2.0, como redes sociais, podem afetar a produtividade na pesquisa qualitativa ao se trabalhar de forma cooperativa.

Palavras-chave: *Computer Assisted Qualitative Data Analysis Software*. Sistemas Colaborativos. Análise Qualitativa de Dados. Redes Sociais.

ABSTRACT

The importance of collaborative environments in a globalized world to enable the sharing of information or make the interaction among people from different locations possible is undeniable. It is also fact, in today's society, the need of carrying out tasks and solving problems in a collaborative way. As an example, one can cite the qualitative research which, when performed with the aid of computers, can make use of CAQDAS (Computer Assisted Qualitative Data Analysis Software) applications. Although it is possible to perform qualitative analysis within CAQDAS applications in isolation, the study/work performed by a group has the differential to enable the interaction among members of a team and provide the expression of different points of view and opinions, besides being more likely to comments and criticisms that help improve the quality of the work. Thus, this document presents WebQDA, a collaborative tool that uses the basic features of qualitative data analysis with the aim of illustrating how Web 2.0 new concepts can affect productivity in qualitative research by working in a cooperative way.

Keywords: Computer Assisted Qualitative Data Analysis Software. Collaborative Systems. Qualitative Data Analysis. Social Networks.

LISTA DE FIGURAS

Figura 1 - Modelo 3C de Colaboração.	31
Figura 2 - Classificação de sistemas colaborativos segundo a abordagem de tempo e espaço.	32
Figura 3 - Classificação de sistemas colaborativos segundo a abordagem das dimensões do Modelo 3C de Colaboração.	33
Figura 4 - Modelo conceitual do WebQDA	40
Figura 5 - Arquitetura conceitual do SCollab Core.....	41
Figura 6 - Arquitetura conceitual do WebQDA	41
Figura 7 - Implementação do método para comunicação com o Twitter	42
Figura 8 - Implementação do método que insere/atualiza um código.....	43
Figura 9 - Mensagem de atualização no Twitter indicando a criação de um novo código no WebQDA.....	44
Figura 10 - Estrutura de projeto do WebQDA	45
Figura 11 - Arquivo remoting-config.xml.....	45
Figura 12 - Criação de um objeto remoto no Flex.....	46
Figura 13 - Tela de login do WebQDA	81
Figura 14 - <i>Workspace</i> do WebQDA	82
Figura 15 - Novo projeto	83
Figura 16 - Listar projetos	84
Figura 17 – Novo código	85
Figura 18 - Listar códigos.....	86

LISTA DE TABELAS

Tabela 1 - Comparativo entre ATLAS.ti e Logos	26
Tabela 2 – Principais requisitos funcionais	39
Tabela 3 - Requisitos funcionais (incluindo proposta de trabalhos futuros).....	59

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CAQDAS	<i>Computer Assisted Qualitative Data Analysis Software</i>
CAT	<i>Coding Analysis Toolkit</i>
COPASEMS	Conselho Paraibano dos Secretários Municipais de Saúde
CSCL	<i>Computer-Supported Cooperative Learning</i>
CSCW	<i>Computer-Supported Cooperative Work</i>
CSSN	<i>Computer-Supported Social Networks</i>
JSF	<i>JavaServer Faces</i>
MVC	<i>Model-view-controller</i>
NESC	Núcleo de Estudos em Saúde Coletiva
RIA	<i>Rich Internet Application</i>
SUS	Sistema Único de Saúde
UFCE	Universidade Federal do Ceará
UFES	Universidade Federal do Espírito Santo
UFPB	Universidade Federal da Paraíba
UNIRIO	Universidade Federal do Estado do Rio de Janeiro

SUMÁRIO

1. Introdução.....	11
1.1. Identificação do problema.....	16
1.2. Objetivos.....	20
1.3. Justificativa.....	20
1.4. Metodologia.....	22
1.5. Organização do trabalho.....	22
2. Ferramentas analisadas.....	24
2.1. Logos.....	24
2.2. ATLAS.ti.....	24
2.3. Comparativo entre os softwares.....	25
3. Fundamentação teórica.....	27
3.1. Análise qualitativa.....	27
3.2. Sistemas colaborativos.....	28
3.2.1. Colaboração.....	30
3.2.2. Sistemas colaborativos e classificação.....	31
3.2.3. Ambientes colaborativos e suas ferramentas.....	33
3.2.4. Redes sociais.....	35
3.2.5. Considerações.....	36
4. A aplicação proposta WebQDA.....	37
4.1. Requisitos levantados.....	38
4.2. Modelo e arquitetura conceituais.....	39
4.3. Colaboração no WebQDA.....	42
4.4. Aspectos de desenvolvimento do WebQDA.....	44
5. Conclusão.....	47
5.1. Propostas de trabalhos futuros.....	48
REFERÊNCIAS.....	49
APÊNDICE A – FERRAMENTAS ANALISADAS.....	53
APÊNDICE B – REQUISITOS DO WEBQDA.....	57
APÊNDICE C – IMPLEMENTAÇÃO DO WEBQDA.....	60
APÊNDICE D – TELAS DO WEBQDA.....	80

1. Introdução

Desde o início da década de 1980, tem sido observado de forma acentuada o desenvolvimento de programas de computador para auxiliar a análise qualitativa de dados (RETTIE *et al.*, 2008). A sigla CAQDAS (*Computer Assisted Qualitative Data Analysis Software*) é comumente utilizada para fazer referência a estes softwares (GIBBS; FRIESE; MANGABEIRA, 2002). São encontrados hoje vários tipos de CAQDAS, desenvolvidos para o gerenciamento de dados, pesquisa textual ou voltados para a construção de mapas conceituais e construção de teoria¹ baseando-se em atividades de codificação e recuperação (*code-and-retrieve*) (LAGE; GODOY, 2008). A tarefa de codificar e recuperar permite que o pesquisador codifique manualmente trechos de dados de acordo com o tema que é objeto de estudo de sua pesquisa, de modo que tal codificação possa então ser recuperada e visualizada separadamente dos dados originais (HOOVER; KOERBER, 2009). As ferramentas mais frequentemente citadas na literatura incluem programas como ATLAS.ti, NUD*IST/NVivo e HyperRESEARCH, todos voltados para a construção de teoria sobre os dados (KELLE, 1997; DOLAN; AYLAND, 2001).

Uma estratégia de grande relevância em todas as metodologias de pesquisa qualitativa é analisar textos provenientes de entrevistas, documentação de interações face a face ou escritas e, embora em menor grau, dados provenientes de imagens visuais, a fim de entender a construção de processos e significados ou acepções que incorporam este tipo de pesquisa (BERGER, 2008). Tal análise pode ser feita manualmente através de métodos tradicionais ou com o uso de software. Em seu trabalho, a autora cita que alguns até consideram esses softwares como sendo muito mais que simples ferramentas para codificação e recuperação, mas como ambientes virtuais complexos para a construção de conhecimento (KONOPÁSEK, 2008), provendo os meios para efetivamente armazenar, gerenciar, classificar, organizar, vincular e comentar as fontes de dados.

Azevedo (1998) atenta para o fato de que, muito embora a designação mais comumente utilizada para estes programas seja a de “software para análise qualitativa de dados”, os mecanismos oferecidos bem como os procedimentos que eles efetuam não são, de fato, analíticos. Os recursos oferecidos por ferramentas CAQDAS são, na verdade, apenas facilitadores de rotinas que são necessárias ao processo analítico. Isto significa que esses programas auxiliam a análise, mas não a fazem pelo pesquisador. O autor defende ainda que o

¹ Onde a teoria é aquilo que “emerge” dos dados (MACMILLAN, 2005).

tipo de auxílio dado à investigação depende, obviamente, do tipo de software que se utiliza, identificando e caracterizando dois tipos básicos de programas:

- a) Recuperadores de texto (*Text Retrievers*): estes programas realizam as tarefas de segmentar os textos, atribuir códigos aos segmentos e, após isso, localizar e apresentar todos os fragmentos associados a uma categoria ou combinação de categorias.
- b) Programas para construção de uma teoria (*Theory Builders*): realizam as mesmas tarefas que os recuperadores de texto, mas possuem funcionalidades específicas que ajudam o pesquisador a desenvolver a teoria. Eles permitem, por exemplo, que sejam estabelecidas conexões entre categorias, a construção de esquemas categoriais com vários níveis, entre outros benefícios.

Lage e Godoy (2008) afirmam que, apesar da variedade de pacotes de software para análise qualitativa, as principais tarefas de gerenciamento e análise de dados realizadas por estas ferramentas são basicamente as mesmas. Segundo as autoras, a maior parte destes softwares é considerada como aplicativos para armazenamento, indexação e recuperação de dados, facilitando a tarefa de atribuir códigos a segmentos de texto e recuperar todos os segmentos que possuem um mesmo código associado. Além destas tarefas, é possível encontrar em pacotes mais sofisticados mecanismos que possibilitam a realização de tarefas mais complexas, como a criação de relacionamentos entre segmentos de dados, por exemplo.

De acordo com Azevedo (1998), é importante esclarecer a relação entre as necessidades do investigador e as características dos vários tipos de software disponíveis para decidir sobre a utilização ou não de uma ferramenta CAQDAS. Ao considerar o uso de um programa para analisar dados qualitativos, alguns aspectos se mostram relevantes para o autor: a) a estruturação e o tipo de dados do projeto; b) o suporte fornecido à codificação; c) as facilidades para elaboração de notas a respeito dos textos ou das codificações; d) as facilidades de busca e apresentação dos dados; e) as facilidades para construir relações entre os dados. Pelo fato de a pesquisa qualitativa ser tão diversa, nenhum pacote de software é “melhor” que os demais. Para Berger (2008), o principal desafio é determinar qual software oferece a melhor solução para as necessidades de um pesquisador específico para um projeto em particular.

Um aspecto de grande relevância presente nessas ferramentas é a forma como os dados são tratados e manuseados, apresentando vantagens no acesso, localização e visualização dos mesmos (BLISMAS; DAINTY, 2003). Além disso, Dolan e Ayland (2001) defendem que esses programas fornecem mecanismos que permitem a realização de tarefas de

armazenamento, gerenciamento e manuseio dos dados com considerável facilidade, possibilitando ao pesquisador trabalhar com quantidades de dados que seriam inviáveis caso fosse utilizada a maneira tradicional (papel, caneta para marcar texto, cola, tesoura, etc.) de análise.

Um aspecto fundamental na análise qualitativa é a codificação. Segundo Azevedo (1998), frequentemente é necessário distribuir os dados por categorias através da indexação de códigos a segmentos textuais, de modo a verificar quais segmentos representam uma determinada categoria. Este processo de categorização pode ser realizado através da atribuição de um ou mais códigos a cada segmento de texto. Vários programas possibilitam a hierarquização de categorias (por exemplo, “Vigilância em Saúde” engloba “Vigilância Epidemiológica”, “Vigilância Ambiental” e “Vigilância Sanitária”) e também oferecem meios de editar a lista completa de códigos utilizados num determinado projeto. O autor expõe ainda que pesquisar o texto baseando-se em palavras-chave que o investigador considera relevantes é um recurso fornecido por todos os programas CAQDAS, sendo possível, assim, localizar palavras, expressões ou frases e combinações destas. Deste modo, “os textos codificados podem ser reagrupados, pesquisados e codificados novamente, facilitando a geração de ideias e novas teorias, num processo de grande proximidade entre o pesquisador e seus dados” (LAGE; GODOY, 2008, p. 80).

Em seu trabalho, Bong (2002) faz referência à codificação como sendo constituída de um processo analítico de quatro passos que consiste em: 1. Comparação de unidades de significado através de categorias para codificação indutiva; 2. Refinamento de categorias; 3. “Delimitação da teoria” através da exploração de relações e padrões entre categorias; 4. Integração de dados para escrever a teoria. A autora considera ainda a análise de dados ao mesmo tempo conceitual e organizacional, interpretativa bem como mecânica, enfatizando que a codificação para recuperação conveniente (de categorias) e construção da teoria (relação entre categorias) envolve a decomposição e dissecação dos dados em unidades analíticas gerenciáveis e significativas. A codificação é, dessa forma, encarada pela autora como um processo de teorização.

Blismas e Dainty (2003) defendem que a tarefa de codificação torna-se muito mais fácil através da utilização de CAQDAS, uma vez que possibilita uma diminuição da quantidade de dados a serem analisados, economizando tempo que pode ser gasto durante o processo de análise. Segundo Johnston (2006), essas ferramentas diminuem o tempo de trabalho necessário e facilitam a análise por possuírem funcionalidades que aproximam o pesquisador dos dados codificados e originais ao mesmo tempo.

Alguns dos principais programas foram projetados baseando-se na *Grounded Theory*² e, como tal, trabalham sistematicamente para construir uma teoria a partir dos dados, o que, em termos práticos, funciona através da criação de categorias hierárquicas e uma aplicação ordenada de códigos ao texto (MACMILLAN, 2005). A autora explica que, após uma primeira codificação, o software é organizado de modo a incentivar uma codificação posterior dos dados em categorias mais precisas, até que o pesquisador tenha o que ele considera as questões que devem passar por uma análise minuciosa.

Dependendo do programa, é possível escrever notas sobre o significado de cada categoria ou sobre outro aspecto qualquer a respeito dos dados ou da codificação. Azevedo (1998) defende que este recurso proporciona a vantagem de permitir a edição de uma espécie de manual de codificação no qual as categorias se apresentam juntamente com os critérios utilizados para a sua classificação. Trata-se de uma facilidade bastante importante quando é preciso realizar o treinamento de pessoas para codificarem algum conjunto de dados qualitativos. Ainda segundo o autor, programas como o NUD*IST e o ATLAS.ti, por exemplo, permitem muito mais do que apenas realizar codificações. Eles possibilitam vincular notas e outras considerações analíticas a segmentos de texto específicos. A ideia aqui é incorporar aspectos da *Grounded Theory* em programas de computador, comenta o autor.

Com relação às ligações estabelecidas entre os dados nos programas para análise qualitativa, as relações podem se processar a vários níveis. Kelle (1997) e Azevedo (1998) explicam que, num nível inicial, uma ligação pode consistir na inclusão de uma determinada categoria numa categorização mais geral ou na subdivisão de uma categoria em subcategorias mais específicas. Num nível mais detalhado, só acessível, portanto, a alguns programas (do tipo *Theory Builders*, segundo os tipos de CAQDAS apresentados no trabalho de Azevedo (1998)), as ligações consistem no estabelecimento de conexões entre vários tipos de dados, tais como notas de campo, categorias, entrevistas, entre outros. Segundo o autor, os programas mais poderosos e eficientes neste sentido possibilitam ao investigador uma fácil “navegação” entre os vários tipos de dados e também visualizar graficamente um esquema que ilustra a relação entre os vários componentes.

² A *Grounded Theory*, ou "teoria fundamentada nos dados" em português, é uma metodologia de pesquisa baseada na análise sistemática de dados. Segundo Yunes e Szymanski (2005, p. 4), a *Grounded Theory* tem como resultado a construção de uma teoria que é "descoberta, desenvolvida e verificada através da coleta e análise de dados" referentes ao fenômeno estudado. O objetivo do pesquisador ao utilizar a *Grounded Theory* é entender uma determinada situação e como e por que seus participantes agem de determinada forma, ou como e por que o fenômeno sob investigação se desdobra desta ou daquela maneira. Após a realização das tarefas que compõem a *Grounded Theory*, o pesquisador acaba então, nas suas conclusões, com algumas teorias que emergiram da sua análise sistemática e rigorosa, ou seja, "os conceitos teóricos emergem dos dados e não são impostos a eles".

Os principais CAQDAS para construção de teoria são dotados de funções avançadas de pesquisa e recuperação de segmentos de texto, possibilitando a realização de buscas tanto sobre os dados originais como sobre os códigos e outros elementos criados pelo pesquisador. Na visão de Lage e Godoy (2008), isto torna possível a percepção de relações entre segmentos de texto e códigos, além de permitir ao pesquisador elaborar correlações entre eles, as quais podem ser utilizadas para a identificação de teorias emergentes. As autoras identificam ainda softwares, como o ATLAS.ti e NVivo, que oferecem recursos para permitir a visualização dessas correlações entre segmentos de dados sob a forma de redes ou mapas conceituais, os quais podem ser armazenados em etapas à medida que a análise se desenvolve.

Segundo Johnston (2006), existia no passado a preocupação de que o uso do computador na análise qualitativa pudesse distanciar o pesquisador de seus dados. Contudo, de acordo com Lage e Godoy (2008), a prática mostra exatamente o contrário, ou seja, a utilização de uma ferramenta CAQDAS muitas vezes acaba aproximando demais o pesquisador de seus dados, fazendo com que o mesmo cometa exageros no processo de codificação, o que poderia interferir no tempo destinado à análise em si. Bong (2002) defende que encarar a codificação como um fim e não como um meio para a construção da teoria é uma armadilha aplicável e prejudicial a ambos os métodos manual e CAQDAS, embora seja mais acentuada neste último, uma vez que o software tem a capacidade de proliferar a codificação. Para a autora, a proliferação de códigos em si não é problemática. O problema está na proliferação de códigos sem (ou independente de) um quadro conceitual. Retirar a codificação de seus pressupostos metodológicos e epistemológicos fornece os seguintes riscos que são variações sobre o excesso de ênfase na codificação (BONG, 2002): a) esquema de codificação mal fundamentado; b) prolongamento do processo de codificação até que o esquema se torne difícil de ser manipulado; c) codificação que assume o controle da análise ao invés de servi-la.

Lage e Godoy (2008) destacam que é importante considerar que as principais ferramentas atuais utilizadas para apoiar a análise qualitativa foram projetadas com base no princípio de que o pesquisador necessita de proximidade e distância de seus dados. De acordo com as autoras, quando é levado em consideração o processo de familiarização com os dados, a proximidade do pesquisador com os mesmos é fornecida através dos mecanismos rápidos de busca, recuperação e retorno ao contexto original. Já a distância é necessária quando se consideram os processos de abstração e síntese, para os quais os recursos oferecidos por essas ferramentas seriam as redes ou mapas conceituais, bem como a verificação de correlações entre códigos e a sumarização facilitada dos dados.

A análise qualitativa de dados e seus processos envolvem a criação de conceitos e a elaboração de tipologias e taxonomias (LAGE; GODOY, 2008). São estes aspectos que constituem o trabalho árduo da análise e que estão a cargo do pesquisador, não importando qual ferramenta CAQDAS está sendo utilizada. Dessa forma, o software não garante por si só a exatidão, a legitimidade e a confiabilidade da análise. Lage e Godoy (2008, p. 84) reforçam ainda que,

ao contrário, o processo de análise, a construção de teorias (caso da *Grounded Theory*), os critérios de codificação e ainda a definição do desenho da pesquisa são atribuições exclusivas do pesquisador, independentemente da ferramenta de apoio, mesmo as mais sofisticadas como NVivo7 e Atlas.ti.

Os principais softwares disponíveis que identificamos em nossa pesquisa para apoiar a análise qualitativa de dados apresentam suas limitações, apesar de alguns serem bastante sofisticados. Como será visto adiante, estes softwares não trabalham em rede. Além disso, não permitem o trabalho em equipe de maneira efetiva, uma vez que o conceito de colaboração presente em alguns desses softwares é muito limitado. Assim, são esses entraves que a presente dissertação pretende atacar, com o intuito de possibilitar uma análise qualitativa mais produtiva através da utilização da Web 2.0.

Uma vez identificados os aspectos e características envolvidos na utilização de software para análise qualitativa, serão apresentados a seguir a identificação do problema a ser investigado, bem como os objetivos deste trabalho e em que contexto o termo CAQDAS se mostra relevante para o estudo.

1.1. Identificação do problema³

O problema investigado que será descrito a seguir está inserido no contexto de um projeto de pesquisa realizado por um grupo de pesquisadores do Núcleo de Estudos em Saúde Coletiva da Universidade Federal da Paraíba (NESC/UFPB).

No Brasil, vêm sendo percebidas evoluções nas políticas que intensificam a participação social em saúde à proporção que o Sistema Único de Saúde – SUS se consolida ao longo dos

³ As informações apresentadas nesta seção foram retiradas da referência Universidade Federal da Paraíba (2009), exceto indicações de outras referências. Tal documento descreve o projeto do Núcleo de Estudos em Saúde Coletiva - NESC apresentado ao Departamento de Apoio à Gestão Participativa DAGEP/SGEP como exigência para solicitação de financiamento ao Ministério da Saúde.

anos. Nessa perspectiva, a prática do controle social se configura como um dos princípios do SUS enquanto mecanismo que possibilitou uma maior participação da sociedade no gerenciamento das políticas públicas. O Ministério da Saúde considera controle social como sendo um processo através do qual a população participa, por meio de representantes, na gestão das políticas públicas do governo, o que abrange a definição, execução e acompanhamento destas. Em outras palavras, controle social é o fenômeno relativo à capacidade da população interferir na gestão pública, direcionando as ações do governo, bem como os seus gastos, de modo que os interesses da coletividade sejam atendidos (ARANTES *et al.*, 2007).

De acordo com as leis que regulamentam o controle social, são estabelecidos dois mecanismos de participação da sociedade na gestão do SUS: as Conferências e os Conselhos de Saúde. Desse modo, por meio de seus representantes, a comunidade “deveria participar da definição das ações de saúde das três esferas de governo, bem como acompanhar e fiscalizar sua execução” (UNIVERSIDADE FEDERAL DA PARAÍBA, 2009, p. 3).

A Paraíba conta com atualmente 223 municípios de modo que todos eles mencionam a existência de Conselhos Municipais de Saúde, organizados de forma paritária. Entretanto, “ainda são incipientes os esforços para avaliar o funcionamento destas instâncias de controle social no estado, particularmente, quanto à sua influência nas políticas públicas de saúde municipais” (UNIVERSIDADE FEDERAL DA PARAÍBA, 2009, p. 3). Um aspecto importante a ser considerado é que a participação da comunidade na gestão das políticas públicas é um fenômeno recente no processo histórico brasileiro e a experiência vem demonstrando quão frágil é esta participação através do surgimento de questões a respeito da atuação social promovida pelas instituições, considerando os entraves de caráter político, cultural e econômico que a participação enfrenta.

Tal constatação mobilizou um grupo de pesquisadores do NESC/UFPB a realizar uma série de estudos a respeito do tema controle social. Estes pesquisadores consideram uma necessidade a obtenção de elementos que auxiliem a compreender como ocorre essa participação e a conhecer os agentes que facilitam ou atravancam a atuação do controle social por meio das Conferências e Conselhos Municipais de Saúde. Dessa forma, a pesquisa empreendida pelo NESC contribui para a efetivação de políticas públicas de saúde, bem como permite o acesso a informações sobre a redistribuição de poder e de recursos, ponto fundamental para gestores e para os movimentos sociais no que diz respeito à consolidação do SUS e à realização do controle social. Vale salientar ainda que o estudo em questão faz parte

de uma pesquisa multicêntrica nacional, cujo objetivo é aprofundar o conhecimento a respeito da participação do controle social na efetivação da política de saúde.

Com relação aos atores e instituições envolvidos na investigação, foram entrevistados prefeitos, secretários municipais do período em estudo e delegados. Também foram ouvidos membros dos Conselhos Municipais de Saúde e do Conselho Paraibano dos Secretários Municipais de Saúde – COPASEMS, além de testemunhas-chave do processo.

No que diz respeito à metodologia utilizada na pesquisa realizada pelo NESC, a investigação teve um enfoque quanti-qualitativo, tendo como objeto de estudo as Conferências Municipais de Saúde. Dessa forma, este trabalho, conforme pode ser observado em seu título, visa dar suporte ao enfoque qualitativo da pesquisa realizada pelo NESC, para o qual foram elaboradas planilhas e um roteiro de entrevista semi-estruturado que serviu para a produção do material empírico. As entrevistas foram empregadas para uma abordagem em profundidade (com prefeitos, secretários municipais, um delegado por segmento da Conferência e testemunhas-chave do processo), sendo adequadas para cada grupo (gestores, profissionais e usuários/familiares).

No tocante aos procedimentos de análise do tipo qualitativa, os documentos e as entrevistas foram analisados sob o enfoque da técnica de análise do discurso (FRASSON, 2007), uma prática especializada em analisar construções ideológicas presentes em um texto. Segundo a autora,

O estudo discursivo considera, em suas análises, não apenas o que é dito em dado momento, mas as relações que esse dito estabelece com o que já foi dito antes e, até mesmo, com o não-dito, atentando, também, para a posição social e histórica dos sujeitos e para as formações discursivas às quais se filiam os discursos (FRASSON, 2007, p.2).

Para a análise do material empírico foi empregado o software ATLAS.ti⁴, ferramenta que fornece suporte à pesquisa qualitativa (CAQDAS). Este software auxilia o gerenciamento de dados, bem como a tarefa de codificação de segmentos de texto através da atribuição de códigos ou categorias de análise identificadas como relevantes para o estudo. No entanto, a utilização de tal ferramenta não permite que os integrantes do projeto trabalhem de forma cooperativa. Alguns CAQDAS, a exemplo do próprio ATLAS.ti, fornecem suporte ao trabalho “colaborativo” através da fusão (*merge*) de subprojetos em um projeto maior, ou seja,

⁴ <http://www.atlasti.com/>

cada integrante é responsável por uma parte do projeto (subprojeto) e, ao término do trabalho, as partes de cada um são unidas de modo a obter o projeto final. Percebe-se, nesse cenário, que o conceito de colaboração é muito restrito, significando apenas a divisão do trabalho necessário para a análise.

Outro problema apresentado por estes pesquisadores está relacionado à codificação das entrevistas analisadas. Por se tratar de um trabalho subjetivo, esse processo de codificação está sujeito a falhas, as quais poderiam ser minimizadas (ou eliminadas) se as ferramentas CAQDAS disponíveis possibilitassem um meio de comunicação entre os integrantes do grupo de pesquisa que pudesse dirimir possíveis dúvidas relacionadas à codificação de segmentos de texto. Para haver esta comunicação, os pesquisadores precisam, obviamente, estar conectados. Daí surge mais um entrave à cooperação na pesquisa qualitativa. Após a realização de pesquisas a respeito de softwares para análise qualitativa, foi identificada até o momento apenas uma ferramenta voltada para a Web (*The Coding Analysis Toolkit - CAT*⁵), sendo as demais aplicações desktop.

Os principais aplicativos CAQDAS disponíveis, incluindo o ATLAS.ti, não podem ser acessados a partir da Web. Ao invés disso, cada pesquisador realiza o seu trabalho em sua própria máquina e as possíveis dúvidas são esclarecidas através de reuniões com o grupo ou outras formas de comunicação. Se estes pesquisadores tivessem à disposição uma ferramenta que minimizasse essas limitações através do uso da Web, o processo de colaboração seria muito mais efetivo.

Por se tratar de um grupo de pesquisa, é inevitável mencionar o termo “rede social”, tão presente ultimamente nas nossas interações através da Internet. Seria interessante se, enquanto um grupo de pesquisadores, o mesmo pudesse também ser encarado como uma rede social. A ideia aqui seria não necessariamente “criar” uma nova rede, mas sim fazer uso das redes sociais já existentes de modo que avanços no processo de comunicação/coordenação do trabalho em equipe pudessem ser alcançados. Por exemplo, seria interessante se houvesse algum mecanismo que informasse, não apenas ao coordenador do projeto, mas a todos os demais integrantes que se relacionassem com ele através de uma rede social, “o que” cada membro da equipe vem fazendo ou que informasse “quando” um integrante está trabalhando no projeto, de modo a auxiliar a coordenação do trabalho em grupo.

⁵ <http://cat.ucsur.pitt.edu/>

1.2. Objetivos

Diante do cenário descrito acima, este trabalho possui como objetivo geral agregar um aspecto mais colaborativo à análise qualitativa de dados. Conforme mencionado anteriormente, o conceito de colaboração presente nas ferramentas CAQDAS atuais é bastante restrito, de modo a abordar a simples divisão de trabalho sem promover uma interação entre os membros de um grupo. Dessa forma, pretende-se ampliar o conceito de “colaboração” agregado aos aplicativos CAQDAS, levando-se em consideração as características e conceitos presentes nos sistemas conhecidos como “colaborativos”, abordados mais adiante.

Como objetivos específicos, pretende-se alcançar os delineados a seguir:

1. Disponibilizar uma ferramenta Web de apoio à análise qualitativa, vencendo as limitações impostas pelos aplicativos desktop atuais no que diz respeito ao acesso a esses softwares;
2. Possibilitar, através da ferramenta proposta, a interação entre o grupo de pesquisadores do NESC, fazendo uso de ferramentas de suporte à colaboração;
3. Fazer uso do conceito de Redes Sociais, de modo a manter o grupo de pesquisa informado a respeito das atividades realizadas por cada membro.
4. Validar o *framework* SCollab Core, proposto por Fujioka (2011) para o desenvolvimento de aplicações sociais para a Web 2.0.

1.3. Justificativa

É inegável a importância dos ambientes colaborativos no mundo globalizado, seja para possibilitar o compartilhamento de informações ou tornar possível a interação entre pessoas distantes. Também é fato, na sociedade atual, a necessidade de realização de tarefas e solução de problemas de forma colaborativa (BRITO; PEREIRA, 2004). Como exemplo, pode-se citar a pesquisa qualitativa, que pode ser auxiliada através da utilização de programas para a análise de dados qualitativos.

Assim, apesar de ser possível realizar análises qualitativas em aplicativos CAQDAS de forma isolada, o estudo/trabalho realizado por um grupo possui o diferencial de permitir a interação entre os membros de uma equipe, possibilitando a expressão de pontos de vista e opiniões diferentes, além de ser mais propenso a comentários e críticas que contribuem para a melhoria e qualidade do trabalho.

Uma questão pouco discutida nas pesquisas relacionadas à utilização de CAQDAS na análise qualitativa é se essas ferramentas facilitam ou não o processo de comunicação entre pesquisadores (LAGE; GODOY, 2008). As autoras afirmam ainda o seguinte:

As principais ferramentas disponíveis trabalham com o conceito de projeto (o Atlas.ti trabalha com o conceito de unidade hermenêutica para os projetos). Cada projeto é formado por um conjunto de dados (transcrições de entrevistas, registros de *focus group* e de observações de campo, teorias, entre outros) e pode ser acessado por apenas um pesquisador ao mesmo tempo. Os principais aplicativos ainda não trabalham em rede. Assim, dois ou mais pesquisadores não podem acessar o mesmo projeto ao mesmo tempo (LAGE; GODOY, 2008, p. 82).

Complementando as ideias das autoras, o que acontece é que pesquisadores geralmente trabalham com cópias iguais do projeto original em suas máquinas individualmente e depois fundem os seus resultados com as funções de *merge* presentes nesses softwares. No entanto, é preciso prestar muita atenção às formas de realizar a fusão entre os elementos do projeto no caso de pesquisas que envolvem grupos de pesquisadores trabalhando em computadores diferentes (PUEBLA, 2003 *apud* LAGE; GODOY, 2008).

Outro aspecto a ser considerado é que o processo de codificação na análise qualitativa exige um esforço analítico do pesquisador no sentido de atribuir códigos (ou categorias) adequados a um determinado segmento de texto. É comum surgirem dúvidas nesse processo, em virtude do mesmo ter um caráter subjetivo, o que pode ocasionar a atribuição de um código inadequado para um fragmento de texto específico. Mais uma vez percebe-se a importância de um meio de comunicação entre os membros de uma equipe trabalhando num mesmo projeto, pois dúvidas relacionadas a esse processo de codificação poderiam ser sanadas através da própria ferramenta utilizada para a análise.

Dessa forma, a implantação de uma ferramenta Web proporciona a comunicação entre pesquisadores de um mesmo projeto, além de auxiliar a coordenação de atividades através das vantagens proporcionadas pelas redes sociais, tornando a análise qualitativa mais produtiva através do uso da Web 2.0. Além disso, as limitações de acesso presentes nas aplicações desktop não mais existirão.

1.4. Metodologia

O primeiro passo para entender as particularidades da análise qualitativa e, conseqüentemente, compreender o problema a ser investigado, foi a realização de videoconferências com pesquisadores da Universidade Federal do Estado do Rio de Janeiro – UNIRIO. Conforme mencionado anteriormente, o estudo realizado pelo NESC faz parte de uma pesquisa multicêntrica nacional, daí a importância do contato estabelecido com os pesquisadores da UNIRIO, uma vez que os mesmos realizam estudos em saúde na mesma linha de pesquisa do NESC. Através das videoconferências realizadas, foi possível identificar aspectos relacionados aos procedimentos efetuados no desenrolar da análise qualitativa, além das dificuldades enfrentadas por estes pesquisadores no que diz respeito às limitações dos softwares utilizados.

A análise de documentações disponíveis no Ministério da Saúde, bem como do documento que descreve e especifica o projeto realizado pelo NESC, contribuiu para a compreensão do contexto no qual a pesquisa do grupo está inserida, possibilitando o entendimento de como foi realizada a abordagem qualitativa do estudo, além do papel dos atores e instituições envolvidos na investigação.

Foi realizada ainda uma capacitação no uso do software ATLAS.ti, por meio da qual foi possível compreender o universo da pesquisa qualitativa em saúde e, também, identificar quais tarefas (ou procedimentos) são necessárias para atender às necessidades destes pesquisadores. Isto direcionou a análise posterior das ferramentas ATLAS.ti e Logos (CAMARGO JÚNIOR, 2000), no sentido de identificar as principais funcionalidades que vão de encontro aos interesses do grupo da UNIRIO e do NESC.

1.5. Organização do trabalho

Este trabalho está organizado da seguinte maneira: o Capítulo 2 apresenta duas ferramentas utilizadas para auxiliar a análise qualitativa de dados, apontando vantagens e desvantagens que foram levadas em consideração para o desenvolvimento da ferramenta proposta; o Capítulo 3 faz um levantamento teórico sobre análise qualitativa de dados e sistemas colaborativos e suas características, explorando o conceito do termo *colaboração* bem como os aspectos relacionados à classificação e às ferramentas deste tipo de sistema, além de destacar a importância dos novos meios de comunicação, como as redes sociais; o Capítulo 4 apresenta o WebQDA e suas características, além dos requisitos levantados para o

desenvolvimento do sistema e seu modelo e arquitetura conceituais; são apresentados ainda neste capítulo aspectos relacionados à colaboração no WebQDA e ao desenvolvimento da ferramenta; no capítulo 5 são apresentadas, finalmente, as considerações a respeito do trabalho.

2. Ferramentas analisadas

Conforme mencionado anteriormente, é importante estar atento para a relação entre as necessidades do pesquisador e as características que os diversos CAQDAS possuem para tomar uma decisão adequada sobre que ferramenta utilizar. Trata-se não exatamente de uma simples “escolha”, mas de uma alternativa a mais adequada possível ao objeto do estudo, aos objetivos e à metodologia que o trabalho requer, bem como aos interesses dos pesquisadores. Desse modo, os softwares ATLAS.ti e Logos são ferramentas que atendem (cada um à sua maneira) às necessidades do grupo de pesquisadores do NESC (no que diz respeito às tarefas para o enfoque qualitativo do estudo), o que motivou uma análise de ambos para identificar as vantagens e desvantagens levadas em consideração para o desenvolvimento do WebQDA, ferramenta apresentada mais adiante.

2.1. Logos

O Logos, software livre desenvolvido por Camargo Júnior (2000), é um gerenciador de dados textuais que fornece suporte na organização e recuperação de informações a partir de textos extensos e não estruturados, como transcrições de entrevistas, diários de campo e outros. O uso do software consiste essencialmente em associar trechos dos documentos trabalhados às categorias de análise presentes na pesquisa e agrupar os diversos trechos identificados, uma tarefa comumente realizada em pesquisas qualitativas. De acordo com o autor, tal processo é bastante interativo e, na prática, pode ser entendido como um trabalho constante de idas e vindas entre os documentos analisados e o material produzido a partir dos mesmos, de modo que haja um refinamento da análise ao longo de seu próprio processo, o que pode despende muito tempo do pesquisador e se transformar em algo excessivamente maçante. Foram estas dificuldades que motivaram o desenvolvimento do software pelo autor.

2.2. ATLAS.ti

O ATLAS.ti é uma ferramenta comercial para análise qualitativa de grandes volumes de dados de texto, gráfico, áudio e vídeo. No decorrer dessa análise, o aplicativo ajuda a explorar o complexo fenômeno escondido nos dados. Este software oferece um ambiente poderoso e intuitivo que mantém o usuário focado no material analisado. São fornecidas ferramentas para

gerenciar, extrair, comparar, explorar e agregar fragmentos significantes de grandes volumes de dados de forma criativa, flexível e sistemática (MUHR; FRIESE, 2004).

2.3. Comparativo entre os softwares

São apresentadas na Tabela 1 a seguir as principais vantagens e desvantagens do Logos e do ATLAS.ti, as quais foram identificadas tendo por base a utilização destes softwares pelo grupo de pesquisadores da UNIRIO, bem como análise própria realizada através de um período de testes utilizando os principais recursos oferecidos por estes aplicativos.

	ATLAS.ti	Logos
Vantagens	<ul style="list-style-type: none"> - Diferentemente do Logos, possui o recurso das Famílias, através do qual são agrupados códigos com a mesma temática. - Possui diversos atalhos na barra lateral (por exemplo, busca de palavra, ir diretamente para uma determinada linha do texto, entre outros). - Permite adicionar comentário ao trecho codificado. - Permite estabelecer relações entre códigos e relações entre segmentos de texto. - A marcação do código fica inserida numa área lateral, de modo que o segmento de texto fica marcado por colchetes do tamanho do trecho selecionado. 	<ul style="list-style-type: none"> - Software livre. - Pode ser utilizado em ambiente Windows ou Linux. - Realiza tabulação por palavras e por códigos.

Desvantagens	- Não existem versões nativas do software para sistemas operacionais que não sejam Windows.	- A marcação do código fica inserida no texto, no início e no final do trecho selecionado.
--------------	---	--

Tabela 1 - Comparativo entre ATLAS.ti e Logos.

Levando-se em consideração as diversas características dos softwares descritos anteriormente, bem como suas vantagens e desvantagens, é importante destacar que a ferramenta proposta neste trabalho não busca competir com o ATLAS.ti ou com o Logos em termos de funcionalidades ou robustez. O objetivo maior é ilustrar como os novos conceitos da Web 2.0, como redes sociais, podem ter impacto e potencial para afetar a produtividade ao se trabalhar de forma colaborativa. Dessa maneira, será utilizado o conceito de redes sociais para auxiliar a coordenação do trabalho em equipe, tornando possível a realização de análises qualitativas através de uma ferramenta disponível via Web, com o intuito de contribuir para a melhoria da pesquisa qualitativa.

3. Fundamentação teórica

Esta seção apresenta um panorama geral sobre análise qualitativa de dados, bem como um estudo a respeito dos principais conceitos abordados neste trabalho, como sistemas colaborativos e redes sociais.

3.1. Análise qualitativa

Faz-se cada vez mais presente hoje uma abordagem que vem se afirmando como uma possibilidade promissora de investigação para diversas áreas do conhecimento: a análise qualitativa. Em linhas gerais, esta abordagem parte de questões ou focos de interesses amplos, que acabam por se definir à proporção que o estudo se desenvolve. A análise qualitativa, dessa maneira, abrange a obtenção de dados descritivos a respeito de pessoas, lugares e processos interativos através do contato direto do pesquisador com a situação estudada, possibilitando a compreensão dos fenômenos envolvidos conforme a perspectiva dos sujeitos/participantes da situação que é objeto de estudo da pesquisa (GODOY, 1995). Apesar da diversidade de trabalhos ditos qualitativos, a autora destaca a seguir alguns aspectos fundamentais que caracterizam e identificam este tipo de estudo:

- a) **A pesquisa qualitativa possui como fonte direta de dados o ambiente natural e o pesquisador é instrumento fundamental:** a principal preocupação na análise qualitativa é o estudo do mundo empírico em seu ambiente natural. Dessa forma, procura-se valorizar o contato direto e prolongado do investigador com a situação que está sendo estudada. Os dados nesta abordagem são frequentemente coletados através da utilização de equipamentos como videoteipes e gravadores ou através de anotações num bloco de papel. Outro aspecto importante é que, para um pesquisador qualitativo, um determinado fenômeno pode ser melhor observado e analisado no contexto em que ocorre e do qual é parte.
- b) **O aspecto descritivo:** Na abordagem qualitativa, os dados coletados aparecem sob a forma de transcrições de entrevistas, fotografias, vídeos, anotações e outros tipos de documentos que descrevem o ambiente que está sendo estudado.
- c) **Preocupação do investigador com o significado que as pessoas dão às coisas e à sua vida:** conforme já exposto, os pesquisadores tentam compreender o ambiente e os fenômenos estudados sob a perspectiva dos sujeitos, considerando todos os pontos de vista como importantes. É preciso, entretanto, assegurar a precisão com

que o pesquisador captou o ponto de vista dos participantes, “testando-o junto aos próprios informantes ou confrontando sua percepção com a de outros pesquisadores” (GODOY, 1995, p.63).

- d) Enfoque indutivo:** Os pesquisadores não partem de hipóteses estabelecidas *a priori*, não havendo a preocupação em buscar evidências que comprovem ou neguem tais suposições. Ao contrário, “quando um pesquisador de orientação qualitativa planeja desenvolver algum tipo de teoria sobre o que está estudando, constrói o quadro teórico aos poucos, à medida que coleta os dados e os examina” (GODOY, 1995, p.63).

Na análise qualitativa, os dados são costumeiramente obtidos de documentos escritos ou documentos orais previamente transcritos, a exemplo de entrevistas e depoimentos. Os dados possuem sentidos que devem ser descobertos através de um processo de segmentação, no qual pedaços de texto relevantes para o estudo do fenômeno em questão (unidades de sentido) são destacados e rotulados com códigos, que são índices de referência adicionados a esses pedaços de texto (MOREIRA, 2000). Tal processo é chamado de codificação e consiste em localizar passagens no material empírico, atribuindo-lhes os significados correspondentes às categorias (códigos) com os quais o investigador está trabalhando. Para Moreira (2000), ainda, tal codificação é o eixo fundamental da pesquisa qualitativa, uma vez que é a partir deste processo que surgirão as explicações e a compreensão do fenômeno estudado.

3.2. Sistemas colaborativos

É natural pensar que com o trabalho colaborativo pode-se obter melhores resultados, pelo menos potencialmente, do que se os integrantes de um grupo atuassem individualmente. Um aspecto interessante em um grupo é o fato de que pode haver uma complementação de habilidades, conhecimentos e esforços, além da interação entre indivíduos que possuem capacidades, entendimentos e pontos de vista complementares. Através da colaboração, os membros de uma equipe podem identificar mais rapidamente possíveis inconsistências e falhas durante o processo de realização de uma tarefa e, conjuntamente, podem buscar ideias e informações que ajudem no processo de resolução de problemas (FUKS; RAPOSO; GEROSA, 2002). O grupo possui ainda maior capacidade para criar alternativas, identificar as vantagens e desvantagens de cada uma delas, solucionar aquelas que são viáveis e, desse modo, tomar decisões. Os autores destacam ainda que, trabalhando em grupo, o integrante de uma equipe se torna mais motivado, uma vez que seu trabalho será analisado e avaliado pelos

demais membros da comunidade da qual ele faz parte. Outro ponto importante apresentado pelos autores é que através da interação com outros integrantes de uma equipe por meio da troca de ideias, o participante acaba trabalhando seus conceitos, refletindo sobre os mesmos e refinando-os.

Nessa perspectiva, um sistema colaborativo é aquele onde um grupo de usuários ou agentes estão engajados numa atividade compartilhada, geralmente a partir de localidades remotas. Em meio à grande variedade de aplicações distribuídas, os sistemas colaborativos são distinguidos pelo fato dos agentes do sistema estarem trabalhando juntos com o intuito de alcançar um objetivo comum e possuem a necessidade de interagir uns com os outros (IVAN; CIUREA, 2009).

Dessa forma, os sistemas colaborativos ganharam importância quando se passou a entender que o ambiente de trabalho exige a realização articulada de tarefas por pessoas que integram uma equipe. Com a evolução dos sistemas colaborativos, definidos ainda como ferramentas de software usadas em redes de computadores com o objetivo de tornar mais fácil a execução de tarefas em grupos, a colaboração em equipes distribuídas geográfica e temporalmente tornou-se possível, tendo a contribuição também da popularização da Internet e dos dispositivos móveis (ALMEIDA; BARANAUSKAS, 2008). Em conformidade com o trabalho dos autores, entre os grupos de sistemas colaborativos que mais ganharam força estão os que compõem as categorias:

- a) *Computer-Supported Cooperative Work (CSCW)*, que abrange a pesquisa para o desenvolvimento de sistemas com foco no trabalho em diversas áreas do conhecimento;
- b) *Computer-Supported Cooperative Learning (CSCL)*, com foco no processo de aprendizagem;
- c) *Computer-Supported Social Networks (CSSN)*, que envolve a pesquisa em sistemas e componentes para redes sociais.

A crescente disseminação do uso de ferramentas computacionais e da Internet foi o grande fator responsável pelo surgimento de uma importante questão: como é que todos esses recursos podem ser aproveitados com o objetivo de permitir, melhor ainda, efetivar a colaboração entre usuários de um determinado grupo? Pesquisas na área de trabalho colaborativo apoiado por computador (*CSCW*, citado anteriormente) vêm sendo desenvolvidas com o intuito de encontrar respostas para tal questionamento. Nesse sentido, o *groupware* (ambiente ou sistema colaborativo – Exemplos: aplicações de bate-papo, blogs, sites de relacionamento, mensagem instantânea, sistemas de recomendação, etc.) surge como a tecnologia que fornece suporte às atividades de pessoas organizadas em grupos (FILIPPO *et*

al., 2007). Em geral, o conceito de *groupware* está relacionado a sistemas especiais que tornam o trabalho numa organização mais fácil, provendo ferramentas de coordenação, comunicação e cooperação através de redes de computadores. Assim, surge uma nova filosofia no desenvolvimento de aplicações. Estas aplicações resolvem um problema particular e, além disso, tornam a comunicação entre os usuários finais mais fácil, assim como a coordenação e cooperação (RUIZ PENICHET *et al.*, 2004).

Para compreender a dinâmica dos ambientes colaborativos, portanto, é importante explorar o conceito do termo *colaboração*, que pode ser estudado com ênfase nos 3Cs (FUKS *et al.*, 2007). Outro ponto de grande valia são os aspectos relacionados aos sistemas colaborativos, sua classificação e ferramentas, além da importância dos novos meios de comunicação, como as redes sociais, que serão discutidos nas seções seguintes.

3.2.1. Colaboração

O conceito do termo *colaboração* está relacionado à ação de trabalhar ou gerar esforços conjuntamente ou, ainda, à realização de determinada tarefa ou trabalho em comum com o apoio de pessoas que constituem um grupo. Contudo, para que a colaboração seja efetivada, determinados aspectos do trabalho em grupo devem ser levados em consideração. O primeiro aspecto a ser considerado é que os integrantes de uma equipe têm a necessidade de trocar informações (Comunicação). Eles necessitam também se organizar (Coordenação) e operar conjuntamente num espaço compartilhado (Cooperação). A análise do contexto que acabou de ser descrito apresenta os elementos constituintes do que é comumente chamado de *Modelo 3C de Colaboração* (FUKS *et al.*, 2007). Os criadores deste modelo caracterizam cada **C** como descrito a seguir.

O **C** de comunicação diz respeito à ação de trocar mensagens com o objetivo de alcançar um entendimento comum das ideias discutidas. A comunicação permite aos integrantes de uma equipe trocar ideias e pontos de vista, argumentando e discutindo com o objetivo de fornecer suporte à realização de negociações, tomada de decisões e também à ação de firmar compromissos.

O segundo **C**, da coordenação, está relacionado à ação de organizar, arranjar, dispor algo sob determinada ordem e método. A atividade de coordenação inclui a identificação de objetivos, o mapeamento destes em tarefas e sua distribuição entre os participantes, sendo essas tarefas realizadas na ordem e tempo previstos e em conformidade com os objetivos e restrições determinados.

Através do C da cooperação, os integrantes de uma equipe produzem, modificam e utilizam de maneira compartilhada uma série de recursos, ou seja, interagem com um conjunto de informações e artefatos. Desse modo, a cooperação trata-se de uma operação conjunta dos integrantes de uma equipe no espaço compartilhado com o objetivo de realizar as tarefas atribuídas a eles, isto é, está relacionada à ação de operar simultaneamente. A Figura 1, extraída de Filippo *et al.* (2007), ilustra a dinâmica entre os 3Cs.



Figura 1 - Modelo 3C de Colaboração.

Um fator importante a ser observado é que, para possibilitar a colaboração, são necessárias informações a respeito do que está acontecendo, que são fornecidas através de elementos de percepção que capturam e condensam as informações coletadas durante a interação entre os integrantes do grupo. Através do conjunto de elementos de percepção disponibilizados pelo ambiente, os integrantes de um grupo conseguem identificar quais etapas já foram finalizadas e quem está responsável pelo cumprimento de determinada tarefa, tornando-os capazes de avaliar seu trabalho e dos demais e realizar adaptações em suas atividades caso seja necessário (FILIPPO *et al.*, 2007).

3.2.2. Sistemas colaborativos e classificação

Enquanto alguns autores utilizam a noção de tempo e espaço para classificar os sistemas colaborativos, existem outros que preferem classificá-los quanto à sua tendência em oferecer suporte a um dos 3Cs (FILIPPO *et al.*, 2007). Tais abordagens são caracterizadas pelos aspectos descritos a seguir.

A classificação segundo a noção de tempo e espaço utiliza dois eixos para representar esses dois elementos. O eixo do tempo é usado para indicar se a interação entre os integrantes

se dá de maneira síncrona ou assíncrona, enquanto que o segundo eixo, o do espaço, serve para indicar se os participantes estão próximos fisicamente ou não. O sistema é classificado como síncrono quando os interlocutores precisam estar conectados simultaneamente ao ambiente para que a colaboração possa ser efetivada. Quando não há necessidade dessa simultaneidade, classifica-se o sistema como assíncrono. A Figura 2, retirada de Filippo *et al.* (2007), ilustra alguns exemplos de sistemas colaborativos classificados segundo essa abordagem, levando em consideração o seu posicionamento nos dois eixos de tempo e espaço.

		TEMPO	
		mesmo tempo (síncrono)	tempo diferente (assíncrono)
ESPAÇO	mesmo lugar (local)	Interações síncronas face-a-face <i>Brainstorming</i>	Interações assíncronas locais <i>Post-It Notes</i>
	local diferente (distribuído)	Interações síncronas distribuídas Bate-papo Videoconferência	Interações assíncronas distribuídas Correio eletrônico Fórum

Figura 2 - Classificação de sistemas colaborativos segundo a abordagem de tempo e espaço.

A segunda abordagem classifica os sistemas colaborativos quanto à sua capacidade de fornecer elementos de um dos 3Cs: Comunicação, Coordenação e Cooperação. Os autores reforçam ainda que, embora um sistema colaborativo seja classificado segundo um dos Cs, não significa que ele não possua elementos dos demais. A Figura 3 (FILIPPO *et al.*, 2007) mostra essa abordagem.



Figura 3 - Classificação de sistemas colaborativos segundo a abordagem das dimensões do Modelo 3C de Colaboração.

3.2.3. Ambientes colaborativos e suas ferramentas

Brito e Pereira (2004) destacam a importância da interação entre pessoas geograficamente distribuídas no sentido da colaboração, além da necessidade de ferramentas que apoiem o trabalho em ambientes colaborativos. Sob a óptica desses autores, com a colaboração, habilidades individuais precisam ser combinadas com as de outras pessoas de modo a produzirem um trabalho de qualidade, pois as ideias passam por um processo de depuração, por meio de críticas e sugestões. Contudo, é necessário ter em mente que, quando se trata de um ambiente virtual, no qual os mecanismos de interação habituais estão ausentes, os processos de interação e decisão são afetados.

Dessa maneira, a **flexibilidade** torna-se um ponto muito importante a ser considerado no desenvolvimento de ambientes colaborativos, podendo ser vista sob duas perspectivas diferentes, a do usuário e a do desenvolvedor (BRITO; PEREIRA, 2004). Em se tratando de usuário, o conceito de flexibilidade abrange fatores como a manipulação de componentes de interface de forma facilitada, o que nos remete ao conceito de usabilidade, a adaptação do ambiente ao seu estilo cognitivo ou, simplesmente, a liberdade de poder colaborar através da interação com diferentes grupos. Já sob a perspectiva do desenvolvedor, o conceito de flexibilidade volta-se ao desempenho, qualidade e facilidade de adição de novos componentes

através do uso de padrões de desenvolvimento que podem ou não ser aplicados pela utilização de frameworks, que fornecem estruturas pré-definidas para a lógica do sistema, diminuindo o tempo de desenvolvimento, embora possam limitar a capacidade de adaptação do sistema a sua estrutura.

Diante disso, é muito importante considerar a forma como ambientes colaborativos são desenvolvidos, bem como novas ferramentas são integradas a esses ambientes, uma vez que muitos deles são isolados e possuem adaptação custosa. Esses sistemas e suas ferramentas devem, portanto, ser elaborados de modo a garantir sua manipulação de maneira facilitada.

Em um ambiente colaborativo, cada ferramenta abrange um dos 3Cs e se integra com um conjunto de outras ferramentas, de modo que os objetivos do usuário possam ser alcançados. Um fator bastante relevante que torna desejável a integração de uma ferramenta com outras é o fato de que uma ferramenta capaz de realizar todo o trabalho do usuário teria um custo de implementação muito elevado, além de não ser capaz de adaptar-se facilmente. Deste modo, para Brito e Pereira (2004, p. 4)

o desenvolvimento de ferramentas de menor granularidade é mais vantajoso, pois ferramentas mais simples podem ser facilmente modificadas e adaptadas, sendo que a capacidade de realização das tarefas de um usuário ocorre pela combinação de diferentes ferramentas.

Ainda na visão dos autores, dependendo do contexto de seu uso, uma ferramenta pode fornecer suporte tanto à comunicação, cooperação ou coordenação, quanto a quaisquer combinações entre os 3Cs. São apresentadas a seguir algumas das ferramentas presentes em ambientes colaborativos descritas pelos autores:

- a) **Bate-papo:** o objetivo desta ferramenta é possibilitar a interação de um usuário com os outros, oferecendo uma comunicação síncrona entre os mesmos.
- b) **Correio eletrônico:** esta ferramenta permite uma comunicação assíncrona através da troca de mensagens de texto e arquivos entre seus usuários.
- c) **Listas de discussão:** através desta ferramenta, vários endereços de correio eletrônico de diferentes usuários são registrados sob um endereço principal. As listas de discussão mantêm o foco em assuntos do interesse de seus integrantes, algumas delas possuindo administradores com a função de decidir censurar ou não mensagens que não estão relacionadas ao assunto discutido na lista.

- d) **Fóruns:** funcionam de modo similar às listas de discussão, permitindo o registro de perguntas e respostas, porém as mensagens são mantidas num local acessado pelos usuários quando os mesmos possuem o interesse em lê-las.
- e) **Mensagens instantâneas:** esta ferramenta permite que seus usuários percebam a presença de colegas no ambiente e comecem uma conversação síncrona, sendo possível que outros usuários sejam convidados a participarem de uma conversação já em andamento.
- f) **Editor de texto compartilhado:** este editor auxilia um grupo de usuários no desenvolvimento de texto em tempo real, sendo o controle de edição similar a uma conferência via voz, de modo que somente um usuário pode editar o texto em um dado momento.

3.2.4. Redes sociais

É importante considerar que as redes sociais estão dentro do modelo 3C de colaboração, uma vez que permitem a comunicação, cooperação e também a coordenação em seus ambientes, estando presentes ainda nas principais aplicações que evoluíram com o advento da Web 2.0 (FUJIOKA, 2011). Como este trabalho procura ilustrar a forma como os novos conceitos da Web 2.0 podem afetar a produtividade ao se trabalhar de forma colaborativa, optou-se por abordar as redes sociais em uma seção à parte.

As redes sociais são uma das formas de representação dos relacionamentos entre seres humanos, sejam afetivos ou de caráter profissional, ou entre grupos de interesses comuns. Esses ambientes funcionam por meio da interação social objetivando a conexão e comunicação entre as pessoas, podendo inclusive ser uma ferramenta muito importante para o desenvolvimento de comunidades científicas (TORQUATO, 2009).

É possível, nesse tipo de ambiente, a criação de comunidades virtuais através das quais as pessoas são conectadas para aprender e criar mais e interagir com pessoas que apresentem interesses afins. As redes sociais priorizam a interação social, a aprendizagem colaborativa e o trabalho cooperativo, demonstrando ser uma importante contribuição para o desenvolvimento da pesquisa científica (ARCOVERDE; MELO; FRANCO, 2008).

Esses ambientes tratam os relacionamentos sociais como nós (ou elos) e ligações entre estes nós, de modo que cada nó representa um ator na rede social e as ligações representam algum tipo de relação entre atores, as quais podem se efetivar de diversas formas, cada uma representando diferentes contextos com relação à rede social. As redes sociais também podem

ser entendidas como grandes repositórios de dados que são responsáveis pelo armazenamento de informações sobre cada usuário que faz parte da rede (COSTA *et al.*, 2008).

3.2.5. Considerações

A forma como o trabalho é realizado, seja em empresas ou instituições, vem sofrendo modificações à medida que se acentua a necessidade de colaboração entre os membros de um grupo, isto é, grande parte do trabalho não está mais sendo realizada de forma individual para a execução de uma determinada tarefa. A intensificação dessa tendência é, pelo menos parcialmente, decorrência do aumento na complexidade das tarefas, que começaram a exigir habilidades multidisciplinares, e do surgimento de novas formas de trabalho, envolvendo vários setores de uma organização ou mesmo várias pessoas durante as fases de elaboração de um produto ou as etapas de desenvolvimento de um projeto.

Com a disseminação das organizações virtuais, das empresas dispersas geograficamente e suas parcerias, a demanda de trabalho colaborativo distribuído aumentou significativamente. Conforme foi observado, a área de pesquisa em CSCW se propõe a estudar o trabalho colaborativo por meio de sistemas computacionais (os *groupwares*) que forneçam apoio à comunicação, coordenação e cooperação (Modelo 3C de colaboração) entre os membros de um grupo, ainda que os mesmos estejam temporal e espacialmente distribuídos.

Foram discutidos nas seções anteriores os principais aspectos relacionados à tecnologia que visa fornecer suporte à realização do trabalho em conjunto através dos sistemas colaborativos. Trata-se de uma área em constante expansão e que engloba uma série de aspectos que vão desde o suporte ao trabalho cooperativo, passando pelo processo de ensino-aprendizagem e a dinâmica das redes sociais, mas que ainda tem muitos campos a serem estudados com minúcia, como a integração de aplicações colaborativas, para que a colaboração possa realmente ser efetivada durante a execução de uma tarefa.

4. A aplicação proposta WebQDA

O WebQDA, cuja designação provém da junção dos termos “Web” (indicando obviamente que se trata de uma aplicação a ser executada na Internet) e “QDA” (acrônimo para *Qualitative Data Analysis*), é uma aplicação que visa auxiliar a análise qualitativa de dados, ou seja, é uma ferramenta CAQDAS (*Computer Assisted Qualitative Data Analysis Software*). A mesma foi projetada para contemplar o conceito de Sistemas Colaborativos, discutido em seções anteriores, permitindo um aumento da produtividade na pesquisa qualitativa através do uso da Web 2.0. O WebQDA foi construído ainda para validar o *framework* SCollab Core (FUJIOKA, 2011), que fornece uma infraestrutura para o desenvolvimento de aplicações sociais.

Conforme já foi mencionado anteriormente, uma estratégia bastante difundida na pesquisa qualitativa é a análise de textos provenientes de entrevistas com a finalidade de compreender a construção de processos e acepções que incorporam o objeto de estudo de um determinado projeto de pesquisa. Neste sentido, a ferramenta em questão visa permitir o gerenciamento de dados e a construção de uma teoria sobre os mesmos através de tarefas de codificação e recuperação. A tarefa de codificar e recuperar no WebQDA possibilita que o pesquisador codifique trechos de entrevistas de acordo com o tema que é objeto de seu estudo, de maneira que esta codificação possa ser recuperada e visualizada separadamente.

A codificação é um aspecto fundamental na análise qualitativa, para o qual a ferramenta proposta apresenta melhorias com relação ao processo realizado pelos pesquisadores do NESC e da UNIRIO ao utilizar o programa ATLAS.ti, por exemplo. Neste software, após realizar o cadastro de categorias e uma posterior segmentação das entrevistas analisadas em citações (ou fragmentos de texto), caso seja necessário realizar alterações nessas categorias, a codificação realizada antes dessas modificações é perdida, inconveniente que não acontece ao se utilizar o programa Logos. Desse modo, procuramos manter esse aspecto positivo do Logos para o processo de codificação e incrementar o aspecto colaborativo à análise qualitativa no WebQDA. Para isto, foram fornecidas à ferramenta características que descrevem o modelo 3C de colaboração proposto por Fuks *et al.* (2007). Assim, são apresentados a seguir os requisitos funcionais do sistema, o modelo de domínio e arquitetura conceitual elaborados, bem como os aspectos de colaboração e de desenvolvimento presentes na ferramenta.

4.1. Requisitos levantados

De acordo com Sommerville (2007), os requisitos de um sistema descrevem as funcionalidades oferecidas bem como suas restrições. Foi realizado o levantamento de algumas questões a respeito das principais funcionalidades da ferramenta proposta, o que contribuiu para a especificação das necessidades de um possível usuário dessa aplicação (no caso, um pesquisador envolvido com análise qualitativa de dados). Para a construção do modelo conceitual do WebQDA, foram levados em consideração os requisitos funcionais descritos no Apêndice B, cujos principais listamos a seguir na Tabela 2.

ID	Requisito	Descrição
RF01	Criar projeto	Este requisito permite que seja criado e armazenado um novo projeto no sistema, que irá armazenar todas as entidades relevantes para a pesquisa. Um projeto no WebQDA receberá um nome que o identifique e poderá conter comentários a seu respeito (recomendado).
RF02	Atribuir documentos a um projeto	Este requisito permite que sejam adicionados/atribuídos documentos (arquivos de texto) a algum projeto armazenado no sistema. Documentos são geralmente criados atribuindo-se arquivos a um projeto. É possível atribuir quantos documentos forem necessários.
RF03	Criar códigos	Este requisito permite que códigos (ou categorias) sejam criados. Códigos capturam significado nos dados. São usados como dispositivos de classificação para criar conjuntos de unidades de informação relacionadas para o propósito da comparação. De uma perspectiva de baixo nível, códigos são tipicamente pequenos pedaços de texto

		referenciando outros pedaços de texto. Seu propósito é classificar um grande número de unidades de dados textuais.
RF04	Criar segmentos de texto (citações)	Este requisito visa criar passagens de texto que são de maior interesse para o pesquisador, atribuindo-lhes um ou mais códigos. Um segmento de texto é um fragmento de um documento primário que é interessante ou importante para o usuário. Em documentos textuais, um segmento é uma seqüência arbitrária de caracteres, variando de um único caractere, uma palavra, uma frase ou um parágrafo até o arquivo inteiro.
RF05	Gerar relatórios	Este requisito visa elaborar um relatório escrito baseado nos códigos, segmentos de texto e memorandos que o pesquisador escreveu ao longo das várias fases do seu projeto. São esses relatórios que permitirão a visualização dos segmentos de texto recuperados após a codificação.

Tabela 2 – Principais requisitos funcionais.

4.2. Modelo e arquitetura conceituais

Um modelo conceitual busca essencialmente reunir todas as entidades e seus relacionamentos, ajudando a definir alguns conceitos de domínios iniciais. A figura a seguir ilustra o modelo conceitual do WebQDA, o qual foi construído com o menor número possível de entidades, visando torná-lo mais simples.

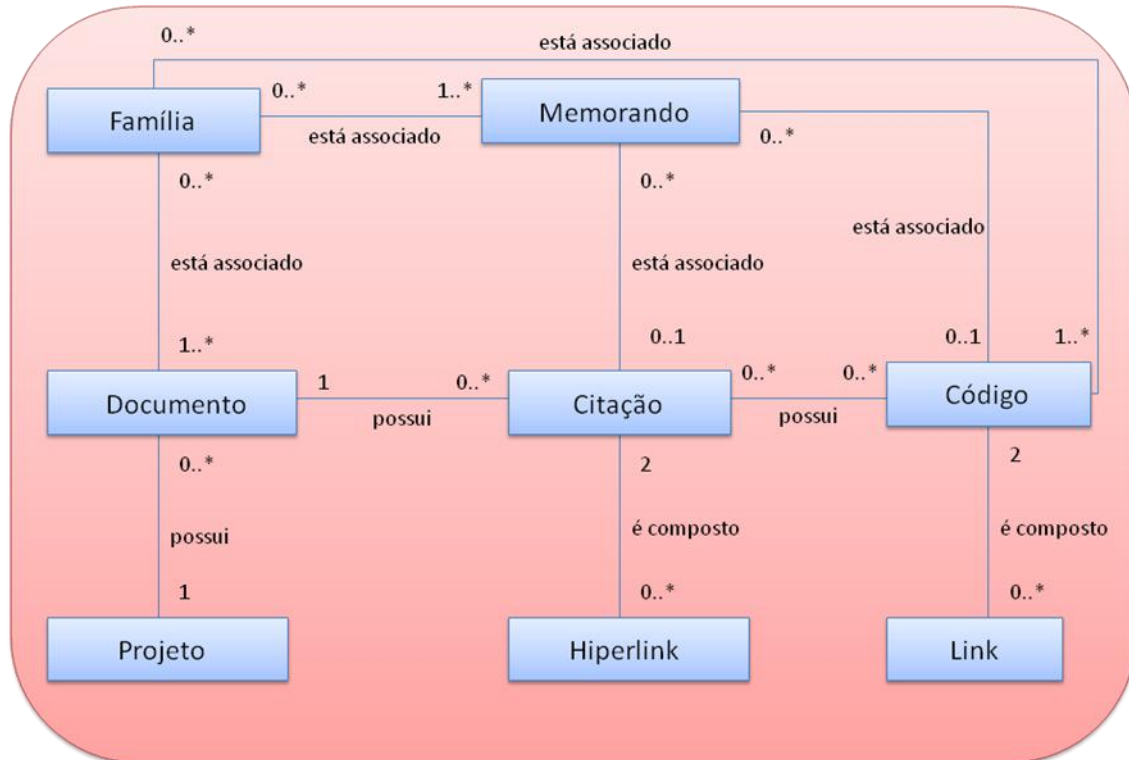


Figura 4 - Modelo conceitual do WebQDA.

O WebQDA trabalha com o conceito de projeto, que irá armazenar todas as entidades relevantes para a pesquisa. Cada projeto possui documentos associados que são as entrevistas que serão analisadas e codificadas. Essas entrevistas são fragmentadas em unidades gerenciáveis de significação chamadas citações, porções do texto que possuem um significado especial para o pesquisador. A essas citações são atribuídos códigos, que são categorias de análise que capturam significado nos dados. Outra entidade presente no modelo conceitual é chamada memorando, através da qual o pesquisador pode realizar anotações a respeito do processo de análise. Documentos, códigos e memorandos podem ser agrupados através do conceito de Família. Além disso, podem ser estabelecidas relações entre códigos (links) e relações entre citações (hiperlinks).

No que diz respeito à arquitetura conceitual, o WebQDA utiliza a proposta por Fujioka (2011) com algumas adaptações. A tecnologia da camada de visão mudou de JSF⁶ (*JavaServer Faces*) para Flex⁷, possibilitando uma maior interatividade com o usuário final. Outra modificação foi o uso do *framework* BlazeDS⁸, responsável por realizar a comunicação

⁶ <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

⁷ <http://www.adobe.com/br/products/flex/>

⁸ <http://opensource.adobe.com/wiki/display/blazeds/BlazeDS/>

do Flex (Visão Web) com a linguagem Java (Aplicação). As Figuras 5 e 6 ilustram essa adaptação.

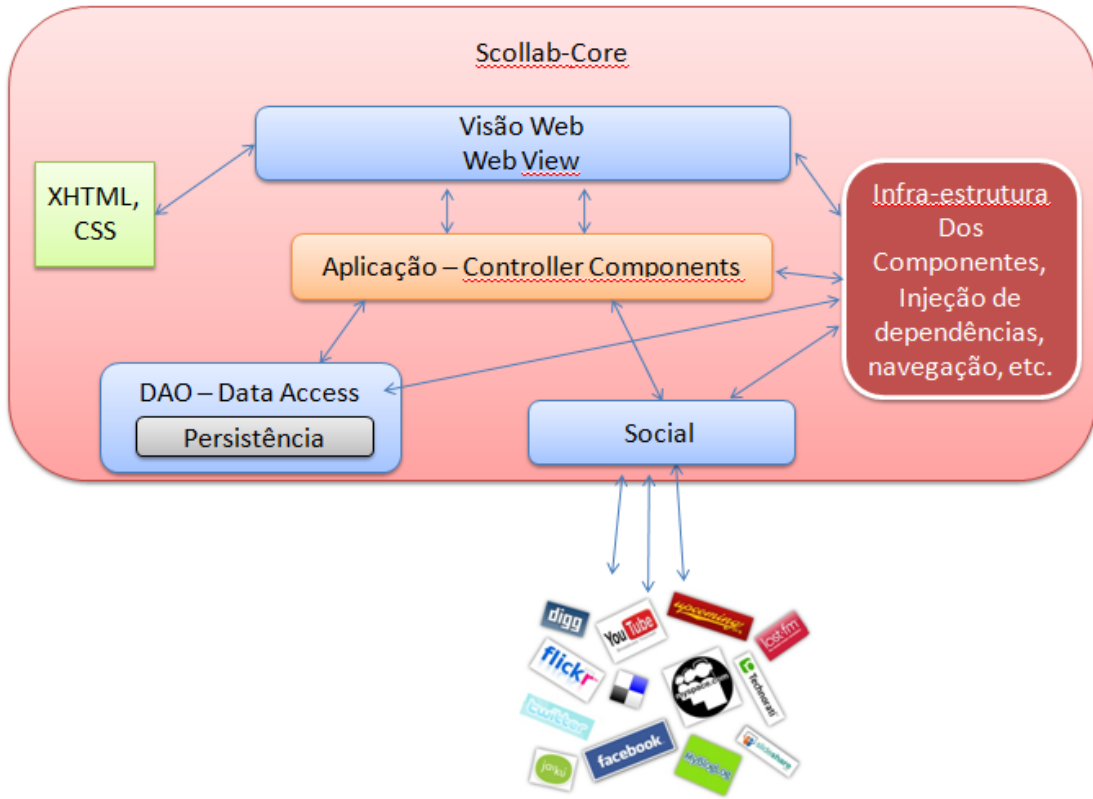


Figura 5 - Arquitetura conceitual do SCollab Core.

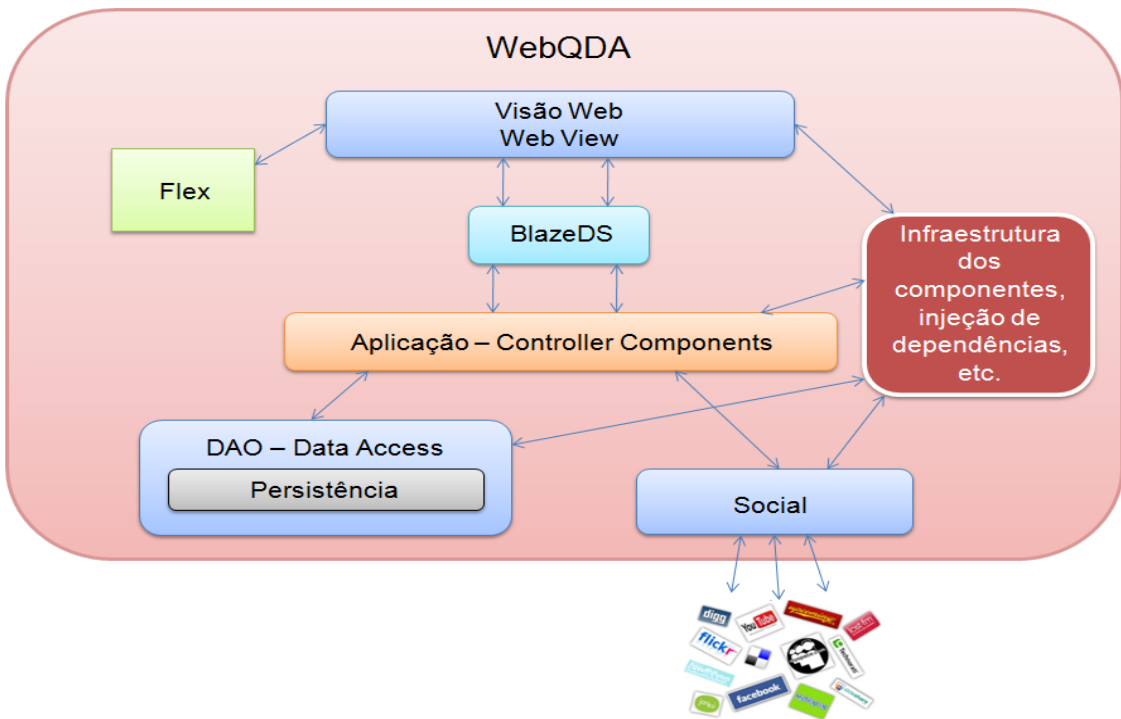


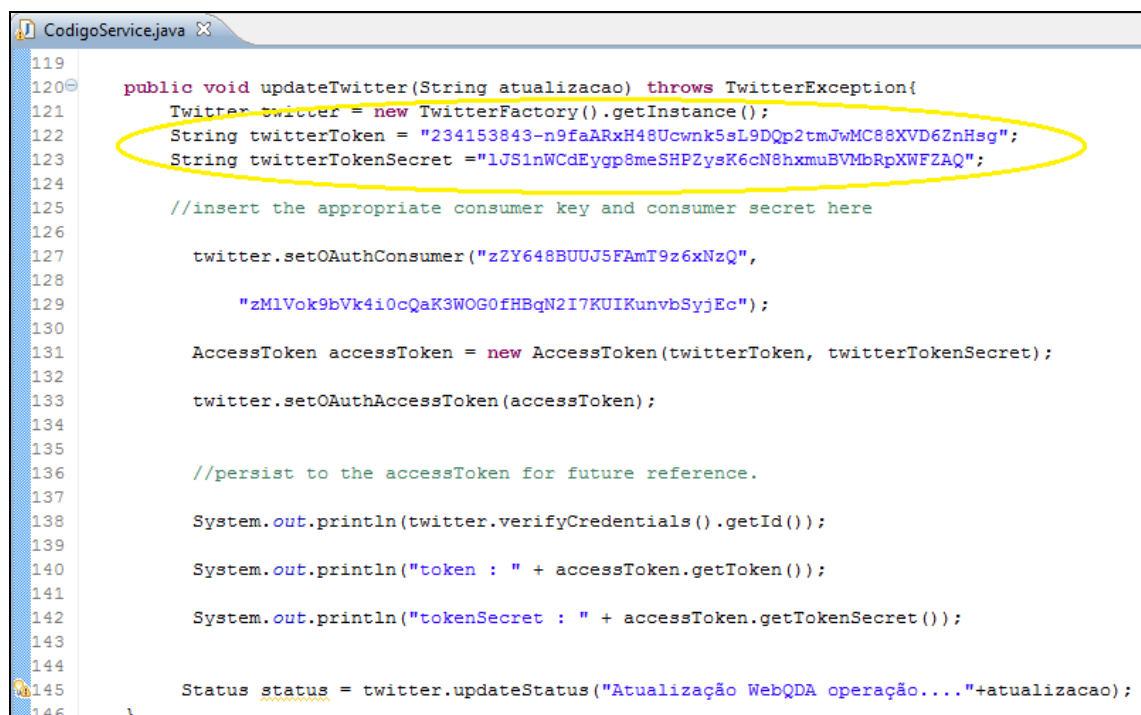
Figura 6 - Arquitetura conceitual do WebQDA.

4.3. Colaboração no WebQDA

Ao utilizar uma ferramenta disponível na Internet, os pesquisadores poderão acessar seus projetos de qualquer lugar, além de fazerem uso dos recursos da Web 2.0 para se comunicarem. Além disso, poderão operar conjuntamente em um ambiente compartilhado, cooperando entre si. Outra característica importante do WebQDA é possibilitar o aproveitamento das vantagens proporcionadas pelas redes sociais para obter avanços no processo de coordenação do trabalho em grupo, além desses ambientes promoverem a comunicação entre seus usuários.

Para o grupo de pesquisadores do NESC e da UNIRIO, é importante a existência de elementos que auxiliem a identificar a participação dos integrantes dos projetos durante o processo de análise realizado por meio de uma ferramenta CAQDAS. Desse modo, no WebQDA os pesquisadores são informados em tempo real do que acontece na ferramenta através de mensagens de atualização enviadas a redes sociais como o Twitter. Isto mostrou a viabilidade do *framework* proposto por Fujioka (2011), permitindo a integração do estudo da análise qualitativa com as redes sociais (no caso com suporte ao Twitter).

Para mostrar como isto é feito, consideremos, por exemplo, a criação de um novo código na ferramenta. A Figura 7 ilustra um método (`updateTwitter`) presente na classe Java `CodigoService` da aplicação, responsável pelas operações relacionadas à persistência de dados.



```

119
120 public void updateTwitter(String atualizacao) throws TwitterException{
121     Twitter twitter = new TwitterFactory().getInstance();
122     String twitterToken = "234153843-n9faARxH48Ucwnk5sL9DQp2tmJwMC88XVD6ZnHsg";
123     String twitterTokenSecret = "1JS1nWCdEygps8meSHPZysK6cN8hxmuBVMbRpXWFZAQ";
124
125     //insert the appropriate consumer key and consumer secret here
126
127     twitter.setOAuthConsumer("zZY648BUUJ5FAMt9z6xNzQ",
128
129         "zM1Vok9bVvk4i0cQaK3WOG0fHBqN2I7KUIKunvbSyjEc");
130
131     AccessToken accessToken = new AccessToken(twitterToken, twitterTokenSecret);
132
133     twitter.setOAuthAccessToken(accessToken);
134
135
136     //persist to the accessToken for future reference.
137
138     System.out.println(twitter.verifyCredentials().getId());
139
140     System.out.println("token : " + accessToken.getToken());
141
142     System.out.println("tokenSecret : " + accessToken.getTokenSecret());
143
144
145     Status status = twitter.updateStatus("Atualização WebQDA operação..." + atualizacao);
146 }

```

Figura 7 - Implementação do método para comunicação com o Twitter.

As variáveis `twitterToken` e `twitterTokenSecret` são tokens de acesso obtidos através de um código de segurança fornecido após autenticação e autorização do WebQDA junto ao Twitter (para detalhes de como esse processo é realizado, sugerimos consulta ao trabalho de Fujioka (2011)). Assim, o método `updateTwitter` é chamado dentro do trecho de código responsável pela inserção/atualização de um código (categoria) na ferramenta, no caso, o método `addUpdateCodigo`, ilustrado na Figura 8.

```

48 public void addUpdateCodigo(Codigo codigo) throws Exception {
49     String str;
50     logger.debug("*** addUpdateCodigo called...");
51     EntityManagerFactory emf = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT);
52     EntityManager em = emf.createEntityManager();
53     // When passing Boolean and Number values from the Flash client to a
54     // Java object, Java interprets null values as the default values for
55     // primitive types; for example, 0 for double, float, long, int, short,
56     // byte.
57     if (codigo.getId() == null || codigo.getId() == 0) {
58         // New code (codigo) is created
59         codigo.setId(null);
60         codigo.setData(new Timestamp(new Date().getTime()));
61         codigo.setDataAtualizacao(new Timestamp(new Date().getTime()));
62     } else {
63         // Existing code (codigo) is updated - do nothing.
64         codigo.setDataAtualizacao(new Timestamp(new Date().getTime()));
65     }
66
67     EntityTransaction tx = em.getTransaction();
68     tx.begin();
69     try {
70         em.merge(codigo);
71         tx.commit();
72         str = "código criado/atualizado "+codigo.getNome();
73         updateTwitter(str);
74     } catch (Exception e) {
75         logger.error("*** Error: " + e.getMessage());
76         tx.rollback();
77         throw new Exception(e.getMessage());
78     } finally {
79         logger.info("*** Closing Entity Manager.");
80         em.close();
81     }
82 }

```

Figura 8 - Implementação do método que insere/atualiza um código.

Dessa maneira, quando um código é inserido/atualizado no WebQDA, é enviada uma mensagem para o Twitter, conforme a Figura 9.



Figura 9 - Mensagem de atualização no Twitter indicando a criação de um novo código no WebQDA.

No exemplo da figura acima, os pesquisadores envolvidos com o trabalho de coleta, análise e classificação dos dados, com o auxílio do WebQDA e a comunicação com as redes sociais como o Twitter, poderão realizar suas atividades sendo avisados sobre eventos em tempo real. Esta possibilidade abre novos horizontes para os agentes interessados no aspecto gerencial (tanto do ponto de vista estatístico como do relacionado à coordenação da capacitação e treinamento das pessoas envolvidas neste processo).

4.4. Aspectos de desenvolvimento do WebQDA

Após levantamento de requisitos através de videoconferências com pesquisadores da UNIRIO e reuniões presenciais com pesquisadores do NESC, além da análise das ferramentas apresentadas anteriormente, deu-se início às atividades de desenvolvimento. O WebQDA foi desenvolvido na linguagem Java, na versão *Enterprise Edition* (Java EE), utilizando-se o padrão *Model-View-Controller* (MVC). Foram aplicados na camada de visão conceitos de *Rich Internet Applications*⁹ (RIA) através da tecnologia Adobe Flex, obtendo maiores graus de interatividade com os usuários. São apresentadas algumas telas da aplicação no Apêndice D deste documento.

A ferramenta apresenta a estrutura de projeto mostrada na Figura 10. Trata-se de um projeto Flex com suporte à integração desta tecnologia da camada de visão com a linguagem

⁹ http://en.wikipedia.org/wiki/Rich_Internet_application
http://www.adobe.com/resources/business/rich_internet_apps/#open

de programação Java. Para tornar possível esta integração, foi necessária a utilização do *framework* BlazeDS, cujas APIs (*Application Programming Interface*) estão localizadas no diretório WebContent\WEB-INF\lib do projeto. Assim, cada classe Java (pacote src_java) possui uma classe *Action Script* na parte Flex do projeto (pacote src_flex), de modo que a comunicação do Flex com Java é realizada através do uso de objetos remotos, conforme especificado no arquivo remoting-config.xml do projeto (Figura 11). Para mais detalhes sobre a implementação do WebQDA, sugerimos consulta ao Apêndice C.

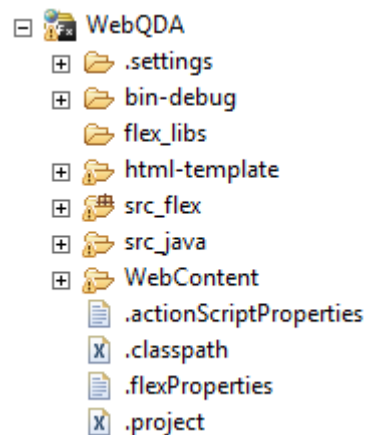


Figura 10 - Estrutura de projeto do WebQDA.


```

remoting-config.xml
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <service id="remoting-service" class="flex.messaging.services.RemotingService">
3
4   <adapters>
5     <adapter-definition id="java-object" class="flex.messaging.services.remoting.adapters.JavaAdapter" default="true" />
6   </adapters>
7
8   <default-channels>
9     <channel ref="my-amf" />
10  </default-channels>
11
12  <!-- WebQDA application -->
13  <destination id="projetoService">
14    <properties>
15      <source>br.ufpb.di.webqda.ProjetoService</source>
16    </properties>
17  </destination>
18
19  <destination id="codigoService">
20    <properties>
21      <source>br.ufpb.di.webqda.CodigoService</source>
22    </properties>
23  </destination>

```

Figura 11 - Arquivo remoting-config.xml.

Com relação à comunicação realizada através de objetos remotos, no arquivo `remoting-config.xml` cada `tag destination` possui uma classe Java associada, de modo que a criação desses objetos na parte Flex é feita de acordo com a Figura 12.



```

6   borderColor="#9AC2FF"
7   defaultButton="{submitButton}"
8   showCloseButton="true"
9   creationComplete="creationCompleteHandler();"
10  close="PopUpManager.removePopUp(this);"
11  title="Novo código"
12  xmlns:text="flash.text.*"
13  height="294"
14  width="506">
15
16  <mx:RemoteObject id="novoCodigo" destination="codigoService" result="handleSaveResult(event)" />
17
18  <vo:Codigo id="codigo"
19      nome="{nomeTextInput.text}"
20      comentario="{comentarioTextInput.text}"/>
21
22  <mx:Script>
23      <![CDATA[
24          import mx.managers.PopUpManager;

```

Figura 12 - Criação de um objeto remoto no Flex.

Com relação à integração do WebQDA com as redes sociais, APIs que fazem a comunicação com essas redes foram importadas, conforme explicitado no trabalho de Fujioka (2011), o *framework* SCollab-Core, que fornece uma infraestrutura para o desenvolvimento de novas aplicações a partir do modelo implementado no próprio *framework* ou possibilita a utilização de apenas seu módulo social, que permite o envio e recebimento de mensagens através de redes sociais como o Facebook¹⁰ e o Twitter¹¹. A utilização do SCollab Core proporcionou o reuso de ideias e artefatos de software para tornar mais rápido o processo de desenvolvimento do WebQDA.

¹⁰ <http://www.facebook.com/>

¹¹ <http://www.twitter.com/>

5. Conclusão

Não se pode negar a importância dos ambientes colaborativos no mundo globalizado, seja para possibilitar o compartilhamento de informações ou tornar possível a interação entre pessoas distribuídas geograficamente e temporalmente. Também é fato, na sociedade atual, a necessidade de realização de tarefas e solução de problemas de modo colaborativo.

Embora análises qualitativas em aplicativos CAQDAS possam ser realizadas de forma isolada, o trabalho realizado por um grupo possui o diferencial de permitir a interação entre os membros de uma equipe, possibilitando a expressão de pontos de vista e opiniões diferentes, contribuindo para a melhoria e qualidade do trabalho.

Neste contexto, o presente estudo mostrou como o conceito de colaboração presente nas principais ferramentas CAQDAS atuais é restrito. Para trabalhar em equipe, os pesquisadores qualitativos utilizam uma cópia de um mesmo projeto em suas máquinas e, para unir o trabalho de cada membro, fazem uso das funções de *merge* presentes nestes softwares. Desse modo, a colaboração nessas ferramentas significa meramente a divisão de tarefas e uma posterior fusão de resultados.

Assim, com a utilização de uma ferramenta disponível através da Internet, a análise qualitativa pode se dar de maneira mais produtiva através do uso da Web 2.0 por uma série de fatores. Primeiro, a limitação de acesso presente nas aplicações desktop não mais existirá, uma vez que os pesquisadores poderão acessar seus projetos de qualquer lugar, bastando para isso apenas estarem conectados à Internet. Segundo, os pesquisadores qualitativos poderão trabalhar conjuntamente, realizando esforços para interagir uns com os outros em um ambiente compartilhado, ou seja, eles poderão “cooperar” entre si. O uso de conceitos da Web 2.0, como as redes sociais, também contribuiu significativamente para uma análise qualitativa mais colaborativa. Além de serem ambientes que promovem a “comunicação” entre os seus usuários, as redes sociais são utilizadas também no WebQDA para o propósito da “coordenação” do trabalho em equipe, identificando quais tarefas estão sendo executadas por determinado usuário. Dessa maneira, o cenário que acabou de ser descrito reforça o modelo 3C de colaboração proposto por Fuks *et al.* (2007).

O WebQDA validou o *framework* SCollab Core proposto por Fujioka (2011). A experiência vivida no Laboratório de Engenharia de Software (MEDEIROS, 2009) mostrou, com nossa aplicação, a importância do reuso de ideias e artefatos de software (*frameworks*) para incrementar o processo de desenvolvimento de aplicações sociais para a Web 2.0.

5.1. Propostas de trabalhos futuros

Como trabalhos futuros, sugerimos a utilização do conceito de Famílias tanto para documentos quanto para códigos. Uma vez que é bastante comum os pesquisadores do NESC e da UNIRIO agruparem as entrevistas semi-estruturadas por município, a presença deste recurso no WebQDA ajudaria na organização dos documentos a serem analisados, onde cada município representaria uma família de documentos. O recurso de Famílias teria também um impacto importante na organização de categorias ou códigos, os quais poderiam ser agrupados de acordo com a temática à qual estivessem relacionados. Conforme exposto no início deste trabalho, os códigos “Vigilância Epidemiológica”, “Vigilância Ambiental” e “Vigilância Sanitária” poderiam fazer parte da Família de códigos “Vigilância em Saúde”, por exemplo, proporcionando uma hierarquização de categorias de análise.

Outra proposta é permitir a criação de relações entre códigos e citações, favorecendo a criação de teorias emergentes a partir dos dados analisados pelos pesquisadores qualitativos. Esta abordagem remete aos CAQDAS classificados como *Theory Builders* ou programas para construção de teorias, que executam as mesmas tarefas que os programas recuperadores de texto, mas são dotados de funcionalidades específicas que auxiliam o pesquisador no desenvolvimento da teoria. No ATLAS.ti, por exemplo, as relações entre citações são chamadas *hiperlinks*, enquanto as relações entre códigos são chamadas *links*.

Do ponto de vista social, aspectos de gerência de informações estatísticas e informações sobre o processo de ensino e aprendizagem da metodologia qualitativa ou abordagem utilizadas poderiam utilizar redes sociais específicas (como YouTube¹², por exemplo) ou novas funcionalidades no WebQDA para consolidar os experimentos vivenciados na pesquisa qualitativa.

¹² <http://www.youtube.com/>

REFERÊNCIAS

ALMEIDA, L. D. A.; BARANAUSKAS, M. C. C. **Um Prospecto de Sistemas Colaborativos: Modelos e Frameworks.** In Proc. VIII Simpósio Sobre Fatores Humanos em Sistemas Computacionais, 21-24 Outubro 2008, 204-213.

ARANTES, Cássia Irene Spinelli *et al.* **O controle social no Sistema Único de Saúde: concepções e ações de enfermeiras da atenção básica.** *Texto & Contexto Enfermagem*, Florianópolis, 2007 jul-set; 16(3): 470-8.

ARCOVERDE, D. F.; MELO, C. A.; FRANCO, R. O. S. **Labbos: uma rede social para pesquisa colaborativa.** IHC 2008 | Demonstrações e Pôsteres. 21-24 Outubro, Porto Alegre – RS, Brasil. Páginas: 326-327.

AZEVEDO, J. **Programas de Computadores para a Análise de Dados Qualitativos.** In A. Esteves & J. Azevedo (eds.) - *Metodologias Qualitativas para as Ciências Sociais*, Porto, Instituto de Sociologia, 1998.

BERGER, R. **Review:** Ann Lewins & Christina Silver (2007). Using Software in Qualitative Research: A Step-by-Step Guide [18 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 10(1), Art. 23, 2008, <http://nbn-resolving.de/urn:nbn:de:0114-fqs0901235>.

BLISMAS, N.; DAINTY, A. **Computer-aided qualitative data analysis: panacea or paradox?** *Building Research & Information*, v. 31, n. 6, p. 455-463, Nov. 2003.

BONG, S. A. **Debunking Myths in Qualitative Data Analysis** [44 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 3(2), Art. 10, 2002, <http://nbn-resolving.de/urn:nbn:de:0114-fqs0202107>.

BRITO, R. F.; PEREIRA, A. T. C. **Um Estudo para Ambientes Colaborativos e suas Ferramentas.** CONAHPA – Congresso Nacional de Ambientes Hipermídia para Aprendizagem. Florianópolis, SC, junho de 2004.

CAMARGO JUNIOR, K. R. **Apresentando logos: um gerenciador de dados textuais.** *Cadernos de Saúde Pública*, Rio de Janeiro, v. 16, n. 1, p. 286-287, 2000.

COSTA, R. *et al.* **A.M.I.G.O.S:** Uma plataforma para Gestão de Conhecimento através de Redes Sociais. Simpósio Brasileiro de Sistemas Colaborativos, 2008. IEEE Xplore, p. 192-203.

DOLAN, A.; AYLAND, C. **Analysis on trial.** *International Journal of Market Research*, v. 43, quarter 4, p. 377-389, 2001.

FILIPPO, D. *et al.* **Ambientes Colaborativos de Realidade Virtual e Aumentada.** In: *Realidade Virtual e Aumentada - Conceitos, Projeto e Aplicações*, Cláudio Kirner e Robson Siscoutto (eds), Editora SBC–Sociedade Brasileira de Computação, Porto Alegre, 2007, ISBN 85-7669-108-6, Cap. 9, pp 168-191.

FRASSON, Carla Beatriz. **Análise do discurso:** considerações básicas. *Cadernos da FUCAMP*, Revista 6, v. 6, jan-dez/2007.

FUJIOKA, Rodrigo da Cruz. **SCollab Core:** Um Framework para Desenvolvimento e Integração de Aplicações JEE a Redes Sociais. João Pessoa, UFPB, 2011. Dissertação (mestrado) – UFPB/CCEN.

FUKS, H. *et al.* **The 3C Collaboration Model.** *The Encyclopedia of E-Collaboration*, Ned Kock (org), ISBN 978-1-59904-000-4, pp. 637-644, 2007.

FUKS, H.; RAPOSO, A. B.; GEROSA, M. A. **Engenharia de Groupware:** Desenvolvimento de Aplicações Colaborativas. XXI Jornada de Atualização em Informática, Anais do XXII Congresso da Sociedade Brasileira de Computação, V2, Cap.3, 2002, ISBN 85-88442-24-8, pp.89-128.

GIBBS, G. R.; FRIESE, S.; MANGABEIRA, W. C. **The Use of New Technology in Qualitative Research. Introduction to Issue 3(2) of FQS** [35 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 3(2), Art. 8, 2002, <http://nbn-resolving.de/urn:nbn:de:0114-fqs020287>.

GODOY, Arilda Schmidt. **Introdução à pesquisa qualitativa e suas possibilidades.** *Revista de Administração de Empresas*. São Paulo, v.35, n. 2, p. 57-63, mar./abr. 1995.

HOOVER, R. S.; KOERBER, A. L. **Using NVivo to Answer the Challenges of Qualitative Research in Professional Communication:** Benefits and Best Practices Tutorial. *IEEE Transactions on Professional Communication*, Vol. 54, No. 1, 2009.

IVAN, Ion; CIUREA, Cristian. **Quality Characteristics of Collaborative Systems.** 2009 Second International Conferences on Advances in Computer-Human Interactions. IEEE Computer Society, 2009.

JOHNSTON, L. **Software and method:** reflections on teaching and using QSR NVivo in doctoral research. *International Journal of Social Research Methodology*, v. 9, n. 5, p. 379-391, Dec. 2006.

KELLE, U. **Theory Building in Qualitative Research and Computer Programs for the Management of Textual Data.** *Sociological Research Online*, vol. 2, no. 2, 1997. <http://www.socresonline.org.uk/socresonline/2/2/1.html>.

KONOPÁSEK, Z. **Making thinking visible with Atlas.ti:** Computer assisted qualitative analysis as textual practices [62 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 9(2), Art. 12, 2008, <http://nbn-resolving.de/urn:nbn:de:0114-fqs0802124>.

LAGE, M. C.; GODOY, A. S. **O Uso do Computador na Análise de Dados Qualitativos:** Questões Emergentes. *RAM – Revista de Administração Mackenzie*, volume 9, n. 4, edição especial, 2008, p. 75-98.

MACMILLAN, K. **More Than Just Coding?** Evaluating CAQDAS in a Discourse Analysis of News Texts [57 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 6(3), Art. 25, 2005, <http://nbn-resolving.de/urn:nbn:de:0114-fqs0503257>.

MEDEIROS, Álvaro Francisco de Castro. **Procedimentos Automáticos para Gerência de Comunidades no Orkut.** Relatório Técnico, Workshop – Labes, out. 2009.

MOREIRA, Daniel Augusto. **Programas de computador para a análise qualitativa:** o caso do QUALPRO. *Administração On Line*, ISSN 1517-7912, v. 1, n. 2, abr./mai./jun. 2000.

MUHR, Thomas; FRIESE, Susanne. **User's Manual for ATLAS.ti 5.0.** 2nd Edition -Berlin, June 2004. Copyright © 2003-2004 by Thomas Muhr, Scientific Software Development, Berlin. All rights reserved. Co-author: Susanne Friese, Quarc Consulting.

RETTIE, R. *et al.* **CAQDAS:** a supplementary tool for qualitative market research. *Qualitative Market Research: An International Journal*. Vol. 11, No. 1, 2008, pp. 76-88.

RUIZ PENICHET, V. M. *et al.* **Implantation Guide for Collaborative Web-based Systems (IGCWS).** Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04). IEEE Computer Society, 2004.

SOMMERVILLE, I. **Engenharia de Software.** 8a. Edição. São Paulo: Pearson Addison-Wesley, 2007.

TORQUATO, J. R. C. **Ubiproject**: uma infra-estrutura para Redes Sociais de projetos compatível com o OAI-PMH. João Pessoa: UFPB, 2009. Dissertação (mestrado) – UFPB/CCEN.

UNIVERSIDADE FEDERAL DA PARAÍBA. Centro de Ciências da Saúde. Núcleo de Estudos em Saúde Coletiva. **O Controle social na Paraíba e as Conferências Municipais de Saúde**: Um estudo avaliativo (2003-2007). 2009.

YUNES, M. A. M.; SZYMANSKI, H. **Entrevista Reflexiva & Grounded-Theory**: Estratégias Metodológicas para Compreensão da Resiliência em Famílias. Revista Interamericana de Psicologia/Interamerican Journal of Psychology, Vol. 39, Num. 3, 2005.

APÊNDICE A – FERRAMENTAS ANALISADAS

Duas foram as ferramentas utilizadas no experimento envolvendo a UFPB, UNIRIO, UFCE e UFES. A razão da escolha se deu frente à política do SUS para utilizar o Logos como proposta de software livre e o ATLAS.ti como aplicação utilizada por algumas instituições ou organizações não-governamentais. A oportunidade de trabalhar em uma equipe multidisciplinar, envolvendo profissionais da Tecnologia da Informação e da área de saúde já familiarizados com esse software, foi determinante na escolha.

O objetivo maior foi estudar as necessidades e deficiências presentes nestes ambientes que forneceram elementos para incrementar as funcionalidades do WebQDA (vide Apêndice B).

Logos

De forma geral, o Logos pode ser descrito da seguinte maneira: os documentos a serem analisados são armazenados em registros numa tabela, de modo que cada registro contém os seguintes campos:

- Um identificador que é único para cada documento;
- Um título;
- Uma identificação de fonte;
- Uma data de referência;
- Um resumo;
- O texto do documento.

Além da tabela de documentos, é gerada também uma tabela de categorias que serão utilizadas para marcar fragmentos de texto de acordo com os critérios do usuário. Cada categoria é identificada através de um código e seu texto de definição, sendo a tabela organizada alfabeticamente pelos códigos. Outra tabela é utilizada para manter o relacionamento entre as categorias e os documentos, de modo a possibilitar a recuperação destes baseando-se em critérios de pesquisa do usuário (CAMARGO JUNIOR, 2000).

ATLAS.ti

O elemento-chave do ATLAS.ti é o que se chama de *Hermeneutic Unit (HU)*, uma espécie de container que armazena todas as entidades relevantes em um projeto. A HU mantém todos os dados em alcance; carregar um projeto com centenas de arquivos é meramente uma questão de abrir uma única HU. São descritos a seguir, além da *Hermeneutic Unit*, os demais conceitos presentes no software (MUHR, 2004):

- a) *Hermeneutic Unit (HU)*: A *Hermeneutic Unit* (ou Unidade Hermenêutica) fornece a estrutura dos dados para cada projeto no ATLAS.ti. Tudo que é relevante para um projeto particular é parte da HU. Por exemplo, os conceitos que serão apresentados a seguir, tais como *primary documents* representando as fontes de dados, *quotations*, *codes* usados para desenvolver conceitos, links conceituais, *memos*, etc, são todos parte de uma HU. Ao selecionar um único arquivo (ativar uma HU), todo o material associado é então ativado automaticamente. O nível mais baixo de uma HU contém os PDs, seguidos pelas *quotations* como seleções de PDs. *Codes* expressam as categorias/variáveis do desenho teórico-metodológico de um estudo. Eles oferecem a possibilidade de dar significado conceitual às *quotations* (citações). *Memos* são pensamentos, reflexões ou idéias associadas geralmente às *quotations* ou aos *codes*.
- b) *Primary Documents (PD)*: Os *Primary Documents* (ou documentos primários) representam os materiais de texto, gráfico, áudio e/ou vídeo que o pesquisador deseja interpretar. O conteúdo de PDs é geralmente armazenado em arquivos de dados no computador do usuário. PDs são geralmente criados atribuindo-se arquivos a uma HU, à qual é possível atribuir quantos documentos forem necessários.
- c) *Quotations*: Uma *quotation* (passagem/citação) é um segmento de um PD que é interessante ou importante para o usuário ou tem expressão conceitual significativa para compreender o objeto de estudo. Geralmente *quotations* são ressaltadas ou atribuídas como relevantes para análise (criadas) pelo pesquisador. Em documentos textuais, uma *quotation* é uma seqüência arbitrária de caracteres, variando de um único caracter, uma palavra, uma frase ou um parágrafo até o arquivo de dados inteiro.

- d) *Codes*: *Codes* (ou códigos) capturam significado nos dados. São usados como dispositivos de classificação para criar conjuntos de unidades de informação relacionadas para o propósito da comparação. De uma perspectiva de baixo nível, *codes* são tipicamente pequenos pedaços de texto referenciando outros pedaços de texto, gráfico, áudio ou vídeo. Seu propósito é classificar um grande número de unidades de dados de texto ou outros tipos de dados.
- e) *Memos*: *Memos* (ou memorandos) capturam pensamentos/conclusões do pesquisador levando em consideração o texto analisado. Um *memo* é similar a um *code*, mas geralmente contém passagens mais longas de texto, podendo ser autônomo ou referenciar *quotations*, *codes* e outros *memos*. Eles podem ser agrupados de acordo com tipos (teórico, descritivo, etc.), tornando útil sua organização e classificação, e também podem ser incluídos como objetos de análise, bastando para isso atribuí-los como PDs.
- f) *Families*: *Families* (ou Famílias) são uma maneira de formar agrupamentos de *PDs*, *codes* e *memos* para um tratamento mais fácil de grupos dessas entidades.
- g) *Network Views*: São representações gráficas que permitem conceitualizar uma estrutura conectando conjuntos de elementos similares juntos num diagrama visual. Com a ajuda das *Network Views* é possível expressar relações entre *codes*, *quotations* e *memos*. Relações são usadas para criar um link entre dois *codes* ou entre duas *quotations*. Um exemplo é a relação “is-a” (ISA – “é um”), que é frequentemente usada para ligar conceitos de diferentes níveis de abstração (por exemplo, DOG <isa> MAMMAL).

Com relação aos processos envolvidos na utilização do software, há dois principais modos de se trabalhar no ATLAS.ti: o *nível textual* – que inclui atividades como segmentação de arquivos de dados, codificação de passagens de texto, imagem, áudio e vídeo e escrita de *memos* – e o *nível conceitual* – que possui foco nas atividades de construção de modelos tais como ligar códigos com base no relacionamento que eles possuem. Essas duas formas de trabalho são descritas a seguir:

- a) Trabalho no nível textual: atividades de pesquisa a nível textual incluem segmentação de *primary documents* em *quotations*, adição de comentários, codificação de passagens do PD selecionado, além do uso de anotações ou *memos* para facilitar sua recuperação.

- b) Trabalho no nível conceitual: através da codificação e recuperação, o ATLAS.ti permite “conectar” visualmente *quotations* selecionadas, *memos* e *codes* em diagramas que mostram graficamente relações complexas. Este recurso transforma virtualmente a área de trabalho do usuário baseada em texto num “playground” gráfico onde o mesmo pode construir conceitos e teorias baseados nas relações entre *codes*, *quotations* ou *memos*.

APÊNDICE B – REQUISITOS DO WEBQDA

São descritos a seguir os requisitos levados em consideração para o desenvolvimento do WebQDA, incluindo aqueles que servirão como proposta de trabalhos futuros, tais como a utilização de “famílias” e a criação de relações entre citações e relações entre códigos.

ID	Requisito	Descrição
RF01	Criar projeto	Este requisito permite que seja criado e armazenado um novo projeto no sistema, que irá armazenar todas as entidades relevantes para a pesquisa. Um projeto no WebQDA receberá um nome que o identifique e poderá conter comentários a seu respeito (recomendado).
RF02	Atribuir documentos a um projeto	Este requisito permite que sejam adicionados/atribuídos documentos (arquivos de texto) a algum projeto armazenado no sistema. Documentos são geralmente criados atribuindo-se arquivos a um projeto. É possível atribuir quantos documentos forem necessários.
RF03	Criar códigos	Este requisito permite que códigos (ou categorias) sejam criados. Códigos capturam significado nos dados. São usados como dispositivos de classificação para criar conjuntos de unidades de informação relacionadas para o propósito da comparação. De uma perspectiva de baixo nível, códigos são tipicamente pequenos pedaços de texto referenciando outros pedaços de texto. Seu propósito é classificar um grande número de unidades de dados textuais.

RF04	Criar segmentos de texto (citações)	Este requisito visa criar passagens de texto que são de maior interesse para o pesquisador, atribuindo-lhes um ou mais códigos. Um segmento de texto é um fragmento de um documento primário que é interessante ou importante para o usuário. Em documentos textuais, um segmento é uma seqüência arbitrária de caracteres, variando de um único caractere, uma palavra, uma frase ou um parágrafo até o arquivo inteiro.
RF05	Criar memorandos	Este requisito permite a escrita de memorandos que contenham uma reflexão do pesquisador sobre os dados, capturando seus pensamentos/conclusões levando em consideração o texto analisado. Um memorando pode ser autônomo ou pode referenciar outras entidades em um projeto.
RF06	Criar famílias	Este requisito visa criar famílias, que são uma maneira de formar agrupamentos de documentos, códigos e memorandos para um tratamento mais fácil de grupos dessas entidades.
RF07	Criar relações entre códigos (links)	Este requisito permite estabelecer relações entre códigos. Para tal, são necessários um código de origem, um código de destino e um nome para identificar a relação entre eles.
RF08	Criar relações entre segmentos de texto (hiperlinks)	Este requisito permite estabelecer relações entre segmentos de texto. São necessários um segmento de origem, um segmento de destino e um nome para identificar a relação entre eles.
RF09	Gerar relatórios	Este requisito visa elaborar um relatório escrito baseado nos códigos, segmentos de texto e memorandos que o pesquisador escreveu ao longo das várias fases do seu projeto. São esses relatórios

		que permitirão a visualização dos segmentos de texto recuperados após a codificação.
--	--	--

Tabela 3 - Requisitos funcionais (incluindo proposta de trabalhos futuros).

APÊNDICE C – IMPLEMENTAÇÃO DO WEBQDA

São apresentados no presente anexo arquivos referentes à codificação da ferramenta proposta, expondo aspectos do Flex e do Java, bem como da integração dessas duas tecnologias. Esta integração permitiu o desenvolvimento de uma aplicação rica para Internet (*RIA – Rich Internet Application*), possibilitando uma maior interação com o usuário final através da tecnologia Adobe Flex, ao mesmo tempo em que tornou possível a obtenção de um *back-end* confiável através do uso da linguagem de programação Java.

Na parte Java da implementação do WebQDA foram utilizados os conceitos referentes à análise qualitativa de dados, como o conceito de projeto, documentos, códigos, etc., conforme as classes mostradas ao longo deste anexo. Já a parte Flex foi responsável por proporcionar uma camada de apresentação mais atrativa para o usuário, de modo que a comunicação do Flex com o Java se deu com o *framework* BlazeDS através da utilização de objetos remotos.

O arquivo Projeto.java mostrado logo abaixo ilustra o conceito de projeto presente no WebQDA, sendo responsável por armazenar todas as entidades relevantes para a pesquisa durante a análise realizada através da ferramenta. Um projeto no WebQDA possui um nome que o identifique e um comentário (opcional) feito pelo pesquisador com o intuito de descrevê-lo ou fornecer informações sobre o mesmo.

Arquivo Projeto.java

```
package br.ufpb.di.webqda;

import java.sql.Timestamp;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

@Entity
@Table(name = "projeto")
@NamedQueries( {
    @NamedQuery(name = "projeto.findAll", query = "from Projeto"),
    @NamedQuery(name = "projeto.byId", query = "select p from Projeto p
where p.id= :idprojeto") })
public class Projeto {
```

```

@Id
@GeneratedValue(strategy=GenerationType.AUTO)
@Column(name = "idprojeto", nullable = false)
private Integer id;

@Basic
@Column(name = "nome", nullable = false, unique = true)
private String nome;

@Basic
@Column(name = "comentario", nullable = true, unique = false)
private String comentario;

@Basic
@Column(name = "data", nullable = false, unique = false)
private Timestamp data;

@Basic
@Column(name = "data_atualizacao", nullable = false, unique = false)
private Timestamp data_atualizacao;

public Projeto(){

}

public Integer getId() {
    return this.id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getComentario() {
    return comentario;
}

public void setComentario(String comentario) {
    this.comentario = comentario;
}

public Timestamp getData() {
    return data;
}

public void setData(Timestamp data) {
    this.data = data;
}

public Timestamp getDataAtualizacao() {
    return data_atualizacao;
}

```

```

    public void setDataAtualizacao(Timestamp data_atualizacao) {
        this.data_atualizacao = data_atualizacao;
    }
}

```

O arquivo ProjetoService.java é responsável por realizar as operações relacionadas à persistência de dados, tais como a criação/atualização de um projeto, bem como a listagem e remoção de projetos. Está presente nesta classe Java um método (updateTwitter) que faz a comunicação com o Twitter, enviando mensagens de atualização para esta rede social quando da criação/atualização/remoção de algum projeto na ferramenta.

Arquivo ProjetoService.java

```

package br.ufpb.di.webqda;

import java.sql.Timestamp;
import java.util.Date;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;
import javax.persistence.Query;
import org.apache.activemq.store.rapid.RapidTopicMessageStore;
import org.apache.log4j.Logger;
import com.sun.org.apache.xalan.internal.xsltc.compiler.util.RtMethodGenerator;
import twitter4j.Status;
import twitter4j.Twitter;
import twitter4j.TwitterException;
import twitter4j.TwitterFactory;
import twitter4j.http.AccessToken;

public class ProjetoService {

    private static final String PERSISTENCE_UNIT = "webqda_db";

    private static Logger logger =
Logger.getLogger(ProjetoService.class);

    public ProjetoService() {
        super();
    }

    public List<Projeto> getProjetos() {
        logger.debug("** getProjetos called...");

        EntityManagerFactory entityManagerFactory =
Persistence.createEntityManagerFactory(PERSISTENCE_UNIT);

        EntityManager em = entityManagerFactory.createEntityManager();

        Query findAllQuery = em.createNamedQuery("projeto.findAll");

```

```

List<Projeto> projetos = findAllQuery.getResultList();

if (projetos != null)
    logger.debug("*** Found " + projetos.size() + "
records:");

return projetos;
}

public void addUpdateProjeto(Projeto projeto) throws Exception {
String str;

logger.debug("*** addUpdateProjeto called...");

EntityManagerFactory emf =
Persistence.createEntityManagerFactory(PERSISTENCE_UNIT);

EntityManager em = emf.createEntityManager();

if (projeto.getId() == null || projeto.getId() == 0) {
// New project is created
projeto.setId(null);
projeto.setData(new Timestamp(new Date().getTime()));
projeto.setDataAtualizacao(new Timestamp(new
Date().getTime()));
} else {
// Existing project is updated - do nothing.
projeto.setDataAtualizacao(new Timestamp(new
Date().getTime()));
}

EntityTransaction tx = em.getTransaction();
tx.begin();
try {
em.merge(projeto);
tx.commit();
str = "projeto criado/atualizado "+projeto.getNome();
updateTwitter(str);
} catch (Exception e) {
logger.error("*** Error: " + e.getMessage());
tx.rollback();
throw new Exception(e.getMessage());
} finally {
logger.info("*** Closing Entity Manager.");
em.close();
}
}

public void deleteProjeto(Integer id) {
String str;
logger.debug("*** deleteProjeto called...");

EntityManagerFactory emf =
Persistence.createEntityManagerFactory(PERSISTENCE_UNIT);

EntityManager em = emf.createEntityManager();

Query q = em.createNamedQuery("projeto.byId");
q.setParameter("idprojeto", id);
Projeto projeto = (Projeto) q.getSingleResult();

```

```

        if (projeto != null) {
            EntityTransaction tx = em.getTransaction();
            tx.begin();
            try {
                em.remove(projeto);
                tx.commit();
                str = "projeto excluído "+projeto.getNome();
                updateTwitter(str);
            } catch (Exception e) {
                logger.error("** Error: " + e.getMessage());
                tx.rollback();
            } finally {
                logger.info("** Closing Entity Manager.");
                em.close();
            }
        }
    }

    public void updateTwitter(String atualizacao) throws
    TwitterException{
        Twitter twitter = new TwitterFactory().getInstance();
        String twitterToken = "234153843-
n9faARxH48Ucwnk5sL9DQp2tmJwMC88XVD6ZnHsg";
        String twitterTokenSecret
        ="1JS1nWCdEygp8meSHPZysK6cN8hxmuBVMbRpXWFZAQ";

        //insert the appropriate consumer key and consumer secret here

        twitter.setOAuthConsumer("zZY648BUUJ5FAmT9z6xNzQ",
            "zM1Vok9bVvk4i0cQaK3WOG0fHBqN2I7KUIKunvbSyjEc");

        AccessToken accessToken = new AccessToken(twitterToken,
twitterTokenSecret);

        twitter.setOAuthAccessToken(accessToken);

        //persist to the accessToken for future reference.

        System.out.println(twitter.verifyCredentials().getId());

        System.out.println("token : " + accessToken.getToken());

        System.out.println("tokenSecret : " +
accessToken.getTokenSecret());

        Status status = twitter.updateStatus("Atualização WebQDA
operação...."+atualizacao);
    }
}

```

O arquivo `Codigo.java` a seguir ilustra o conceito de código no WebQDA, entidade responsável por capturar sentido nos dados analisados pelo pesquisador. Um código possui um nome que o identifique e um comentário (opcional) que permite que as categorias sejam apresentadas juntamente com os critérios utilizados para sua classificação, ou seja, através deste comentário é possível descrever a categoria (código) em questão.

Arquivo Codigo.java

```

package br.ufpb.di.webqda;

import java.sql.Timestamp;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

@Entity
@Table(name = "codigo")
@NamedQueries( {
    @NamedQuery(name = "codigo.findAll", query = "from Codigo"),
    @NamedQuery(name = "codigo.byId", query = "select p from Codigo p
where p.id= :idcodigo" ) })

public class Codigo {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name = "idcodigo", nullable = false)
    private Integer id;

    @Basic
    @Column(name = "nome", nullable = false, unique = true)
    private String nome;

    @Basic
    @Column(name = "comentario", nullable = true, unique = false)
    private String comentario;

    @Basic
    @Column(name = "familia_idfamilia", nullable = true, unique = false)
    private Integer idfamilia;

    @Basic
    @Column(name = "data", nullable = false, unique = false)
    private Timestamp data;

    @Basic
    @Column(name = "data_atualizacao", nullable = false, unique = false)
    private Timestamp data_atualizacao;

    public Codigo() {

    }

    public Integer getId() {
        return this.id;
    }
}

```

```

public void setId(Integer id) {
    this.id = id;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getComentario() {
    return comentario;
}

public void setComentario(String comentario) {
    this.comentario = comentario;
}

public Integer getIdFamilia() {
    return this.idfamilia;
}

public void setIdFamilia(Integer idfamilia) {
    this.idfamilia = idfamilia;
}

public Timestamp getData() {
    return data;
}

public void setData(Timestamp data) {
    this.data = data;
}

public Timestamp getDataAtualizacao() {
    return data_atualizacao;
}

public void setDataAtualizacao(Timestamp data_atualizacao) {
    this.data_atualizacao = data_atualizacao;
}
}

```

O arquivo `CodigoService.java` é responsável por realizar as operações relacionadas à persistência de dados, tais como a criação/atualização de um código, bem como a listagem e remoção de códigos. Esta classe Java possui um método (`updateTwitter`) que faz a comunicação com o Twitter, enviando mensagens de atualização para esta rede social quando da criação/atualização/remoção de algum código na ferramenta.

Arquivo `CodigoService.java`

```
package br.ufpb.di.webqda;
```

```

import java.sql.Timestamp;
import java.util.Date;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;
import javax.persistence.Query;
import org.apache.log4j.Logger;
import twitter4j.Status;
import twitter4j.Twitter;
import twitter4j.TwitterException;
import twitter4j.TwitterFactory;
import twitter4j.http.AccessToken;

public class CodigoService {

    private static final String PERSISTENCE_UNIT = "webqda_db";

    private static Logger logger = Logger.getLogger(CodigoService.class);

    public CodigoService() {
        super();
    }

    public List<Codigo> getCodigos() {
        logger.debug("** getCodigos called...");

        EntityManagerFactory entityManagerFactory =
Persistence.createEntityManagerFactory(PERSISTENCE_UNIT);

        EntityManager em = entityManagerFactory.createEntityManager();

        Query findAllQuery = em.createNamedQuery("codigo.findAll");
        List<Codigo> codigos = findAllQuery.getResultList();

        if (codigos != null)
            logger.debug("** Found " + codigos.size() + " records:");

        return codigos;
    }

    public void addUpdateCodigo(Codigo codigo) throws Exception {
        String str;

        logger.debug("** addUpdateCodigo called...");

        EntityManagerFactory emf =
Persistence.createEntityManagerFactory(PERSISTENCE_UNIT);

        EntityManager em = emf.createEntityManager();

        // When passing Boolean and Number values from the Flash client
to a
        // Java object, Java interprets null values as the default
values for
        // primitive types; for example, 0 for double, float, long,
int, short,
        // byte.

```

```

        if (codigo.getId() == null || codigo.getId() == 0) {
            // New code (codigo) is created
            codigo.setId(null);
            codigo.setData(new Timestamp(new Date().getTime()));
            codigo.setDataAtualizacao(new Timestamp(new
Date().getTime()));
        } else {
            // Existing code is updated - do nothing.
            codigo.setDataAtualizacao(new Timestamp(new
Date().getTime()));
        }

        EntityTransaction tx = em.getTransaction();
        tx.begin();
        try {
            em.merge(codigo);
            tx.commit();
            str = "código criado/atualizado "+codigo.getNome();
            updateTwitter(str);
        } catch (Exception e) {
            logger.error("** Error: " + e.getMessage());
            tx.rollback();
            throw new Exception(e.getMessage());
        } finally {
            logger.info("** Closing Entity Manager.");
            em.close();
        }
    }

    public void deleteCodigo(Integer id) {
        String str;
        logger.debug("** deleteCodigo called...");

        EntityManagerFactory emf =
Persistence.createEntityManagerFactory(PERSISTENCE_UNIT);

        EntityManager em = emf.createEntityManager();

        Query q = em.createNamedQuery("codigo.byId");
        q.setParameter("idcodigo", id);
        Codigo codigo = (Codigo) q.getSingleResult();

        if (codigo != null) {
            EntityTransaction tx = em.getTransaction();
            tx.begin();
            try {
                em.remove(codigo);
                tx.commit();
                str = "código excluído "+codigo.getNome();
                updateTwitter(str);
            } catch (Exception e) {
                logger.error("** Error: " + e.getMessage());
                tx.rollback();
            } finally {
                logger.info("** Closing Entity Manager.");
                em.close();
            }
        }
    }
}

```

```

    public void updateTwitter(String atualizacao) throws
TwitterException{
    Twitter twitter = new TwitterFactory().getInstance();
    String twitterToken = "234153843-
n9faARxH48Ucwnk5sL9DQp2tmJwMC88XVD6ZnHsg";
    String twitterTokenSecret
="1JS1nWCdEygp8meSHPZysK6cN8hxmuBVMbRpXWFZAQ";

    //insert the appropriate consumer key and consumer secret here

    twitter.setOAuthConsumer("zZY648BUUJ5FAmT9z6xNzQ",

        "zM1Vok9bVvk4i0cQaK3WOG0fHBqN2I7KUIKunvbSyjEc");

    AccessToken accessToken = new AccessToken(twitterToken,
twitterTokenSecret);

    twitter.setOAuthAccessToken(accessToken);

    //persist to the accessToken for future reference.

    System.out.println(twitter.verifyCredentials().getId());

    System.out.println("token : " + accessToken.getToken());

    System.out.println("tokenSecret : " +
accessToken.getTokenSecret());

    Status status = twitter.updateStatus("Atualização WebQDA
operação...."+atualizacao);
    }
}

```

O arquivo NovoProjeto.mxml ilustra uma das telas da camada de apresentação, responsável pela criação de um novo projeto no WebQDA, sendo passados um nome e um comentário para o projeto. É criado neste arquivo um objeto remoto que faz a comunicação do Flex com o Java (classe ProjetoService.java).

Arquivo NovoProjeto.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:TitleWindow
    xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:vo="br.ufpb.di.webqda.vo.*"
    layout="absolute"
    defaultButton="{submitButton}"
    showCloseButton="true"
    creationComplete="creationCompleteHandler();"
    close="PopUpManager.removePopUp(this);"
    title="Novo projeto"
    xmlns:text="flash.text.*"
    borderColor="#9AC2FF"
    height="294"

```

```

width="506">

    <mx:RemoteObject id="novoProjeto" destination="projetoService"
result="handleSaveResult(event)" fault="handleFault(event)"
showBusyCursor="true" />

    <vo:Projeto id="projeto"
        nome="{nomeTextInput.text}"
        comentario="{comentarioTextInput.text}"/>

    <mx:Script>
        <![CDATA[
            import mx.managers.PopUpManager;
            import mx.controls.Alert;
            import mx.containers.Canvas;
            import mx.rpc.events.ResultEvent;
            import mx.rpc.events.FaultEvent;
            import mx.events.ValidationResultEvent;
            import mx.validators.Validator;

            [Bindable]
            private var message:String;

            [Bindable]
            private var formIsValid:Boolean = false;

            [Bindable]
            public var formIsEmpty:Boolean;

            // Holds a reference to the currently focussed control on
the form.
            private var focussedFormControl:DisplayObject;

            private function handleSaveResult(ev:ResultEvent):void {
                clearFormHandler();
                validateForm(ev);
                Alert.show("Projeto criado/atualizado com sucesso.",
"Informação", Alert.OK, null, null, null, Alert.OK);
                parentApplication.listarProjetos.loaderService.getProjeto
s();
                PopUpManager.removePopUp(this);
            }

            private function handleFault(ev:FaultEvent):void {
                message = "Error: " + ev.fault.faultCode + " \n "
                    + "Detail: " + ev.fault.faultDetail + " \n "
                    + "Message: " + ev.fault.faultString;
            }

            public function saveProject():void {
                novoProjeto.addUpdateProjeto(projeto);
            }

            private function creationCompleteHandler():void {
                PopUpManager.centerPopUp(this);
                resetFocus();
            }

            private function resetFocus():void {
                focusManager.setFocus(nomeTextInput);
            }
        ]]>

```

```

// Validate the form
public function validateForm(event:Event):void {
    focussedFormControl = event.target as DisplayObject;
    formIsValid = true;
    formIsEmpty = (nomeTextInput.text == "");
    validate(nomeValidator);
    validate(comentarioValidator);
}

private function validate(validator:Validator):Boolean {
    // See also here for validating data:
    //
http://www.adobe.com/devnet/flex/quickstart/validating\_data/
    var validatorSource:DisplayObject = validator.source as
DisplayObject;
    var suppressEvents:Boolean = (validatorSource !=
focussedFormControl);
    var event:ValidationResultEvent = validator.validate(null,
suppressEvents);
    var currentControlIsValid:Boolean = (event.type ==
ValidationResultEvent.VALID);

    formIsValid = formIsValid && currentControlIsValid;
    return currentControlIsValid;
}

private function clearFormHandler():void {
    // Clear all input fields.
    nomeTextInput.text = "";
    comentarioTextInput.text = "";

    message = "";

    // Clear validation error messages.
    nomeTextInput.errorString = "";
    comentarioTextInput.errorString = "";

    formIsEmpty = true;
    formIsValid = false;
    resetFocus();
}
]]>
</mx:Script>

<mx:StringValidator id="nomeValidator" source="{nomeTextInput}"
property="text" minLength="2" required="true" />
<mx:StringValidator id="comentarioValidator"
source="{comentarioTextInput}" property="text" minLength="2"
required="false" />

<mx:Form width="100%" height="162" >
    <mx:FormItem label="Nome" required="true">
        <mx:TextInput id="nomeTextInput" text="{projeto.nome}"
change="validateForm(event);" width="345"/>
    </mx:FormItem>
    <mx:FormItem label="Comentário" required="false">

```

```

        <mx:TextInput id="comentarioTextInput"
            text="{projeto.comentario}" change="validateForm(event);"
            width="345" height="98"/>
        </mx:FormItem>
    </mx:Form>

    <mx:Spacer height="10" y="121" x="19"/>
    <mx:Text text="{message}" fontWeight="bold" width="300" x="10"
        y="170"/>
    <mx:ControlBar horizontalAlign="center">
        <mx:Button label="Salvar projeto" id="submitButton"
            enabled="{formIsValid}" click="saveProject();" />
        <mx:Button label="Cancelar"

            click="PopUpManager.removePopUp(this);" />
    </mx:ControlBar>

</mx:TitleWindow>

```

O arquivo ListarProjetos.mxml ilustra a tela da aplicação responsável por realizar a listagens de projetos existentes, fornecendo opções para criação/atualização/remoção de projetos.

Arquivo ListarProjetos.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Panel
    xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:view="br.ufpb.di.webqda.view.*"
    layout="absolute"
    borderColor="#9AC2FF"
    title="Projetos - Encontrados {projetoRecords} registros."
    creationComplete="loadProjetos();"
    height="70%"
    width="70%">

    <mx:RemoteObject id="loaderService" destination="projetoService"
        result="handleLoadResult(event)" fault="handleFault(event)"
        showBusyCursor="true" />
    <mx:RemoteObject id="deleteService" destination="projetoService"
        result="handleDeleteResult(event)" fault="handleFault(event)"
        showBusyCursor="true" />

    <mx:Script>
        <![CDATA[
            import br.ufpb.di.webqda.vo.Projeto;
            import mx.controls.Alert;
            import mx.managers.PopUpManager;
            import mx.containers.TitleWindow;
            import mx.collections.ArrayCollection;
            import mx.rpc.events.ResultEvent;
            import mx.rpc.events.FaultEvent;

            [Bindable]
            private var message:String;

```

```

        [Bindable]
        private var projetos:ArrayCollection = new
ArrayCollection();

        [Bindable]
        private var projetoRecords:int = 0;

        public function loadProjetos():void {
            loaderService.getProjetos();
        }

        private function deleteProjeto():void {
            if(dataGrid.selectedItem != null) {
                var selectedItem:Projeto =
dataGrid.selectedItem as Projeto;
                deleteService.deleteProjeto(selectedItem.id);
            }
        }

        private function createProjeto():void {
            var titleWindow:NovoProjeto =
NovoProjeto(PopUpManager.createPopUp(this, NovoProjeto, true));
            titleWindow.setStyle("borderAlpha", 0.9);
            titleWindow.formIsEmpty = true;
        }

        private function updateProjeto():void {
            var titleWindow:NovoProjeto =
NovoProjeto(PopUpManager.createPopUp(this, NovoProjeto, true));
            titleWindow.setStyle("borderAlpha", 0.9);
            titleWindow.projeto = dataGrid.selectedItem as
Projeto;
            titleWindow.formIsEmpty = false;
        }

        private function handleLoadResult(ev:ResultEvent):void {
            projetos = ev.result as ArrayCollection;
            projetoRecords = projetos.length;
        }

        private function handleDeleteResult(ev:ResultEvent):void
{
            Alert.show("Projeto excluído com sucesso.",
"Informação", Alert.OK, null, null, null, Alert.OK);
            loadProjetos();
        }

        private function handleFault(ev:FaultEvent):void {
            message = "Error: "
                + ev.fault.faultCode + " - "
                + ev.fault.faultDetail + " - "
                + ev.fault.faultString;
        }

        private function fecharPanel():void {
            this.visible=false;
            parentApplication.login.appMenuBar.enabled=true;
        }
    ]]>
</mx:Script>

```

```

<mx:VBox width="100%" height="100%">
  <mx:Label text="{message}" fontWeight="bold"
includeInLayout="false" />
  <mx:DataGrid
    id="dataGrid"
    width="100%"
    height="100%"
    alternatingItemColors="[#F7F7F7, #E3F0FF]"
    dataProvider="{projetos}"
    doubleClickEnabled="true"
    doubleClick="updateProjeto()" >
    <mx:columns>
      <mx:DataGridColumn dataField="id"
headerText="ID Projeto" width="100"/>
      <mx:DataGridColumn dataField="nome"
headerText="Nome" />
      <mx:DataGridColumn dataField="comentario"
headerText="Comentário" />
      <mx:DataGridColumn dataField="data"
headerText="Criado" />
      <!-- <mx:DataGridColumn dataField="data_atualizacao"
headerText="Modificado" /> -->
    </mx:columns>
  </mx:DataGrid>
</mx:VBox>
<mx:ControlBar horizontalAlign="center">
  <mx:Button label="Criar projeto"
click="createProjeto()" tooltip="Cria um novo projeto e o
armazena na base de dados." />
  <mx:Button label="Atualizar projeto"
click="updateProjeto()" enabled="{dataGrid.selectedItem}"
tooltip="Atualiza um projeto existente na base de dados." />
  <mx:Button label="Excluir projeto"
click="deleteProjeto()" enabled="{dataGrid.selectedItem}"
tooltip="Exclui o projeto da base de dados." />
  <mx:Button label="Recarregar dados" click="loadProjetos()"
tooltip="Recarrega a lista de projetos da base de dados." />
  <mx:Button label="Fechar"
click="fecharPanel()" tooltip="Fecha este painel." />
</mx:ControlBar>
</mx:Panel>

```

O arquivo NovoCodigo.mxml ilustra mais uma tela da camada de apresentação, responsável pela criação de um novo código no WebQDA, sendo passados um nome e um comentário para o código. É criado também neste arquivo um objeto remoto que faz a comunicação do Flex com o Java (classe CodigoService.java).

Arquivo NovoCodigo.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:TitleWindow
  xmlns:mx="http://www.adobe.com/2006/mxml"
  xmlns:vo="br.ufpb.di.webqda.vo.*"
  layout="absolute"
  borderColor="#9AC2FF"

```

```

defaultButton="{submitButton}"
showCloseButton="true"
creationComplete="creationCompleteHandler();"
close="PopUpManager.removePopUp(this);"
title="Novo código"
xmlns:text="flash.text.*"
height="294"
width="506">

    <mx:RemoteObject id="novoCodigo" destination="codigoService"
result="handleSaveResult(event)" fault="handleFault(event)"
showBusyCursor="true" />

    <vo:Codigo id="codigo"
        nome="{nomeTextInput.text}"
        comentario="{comentarioTextInput.text}" />

    <mx:Script>
        <![CDATA[
            import mx.managers.PopUpManager;
            import mx.controls.Alert;
            import mx.containers.Canvas;
            import mx.rpc.events.ResultEvent;
            import mx.rpc.events.FaultEvent;
            import mx.events.ValidationResultEvent;
            import mx.validators.Validator;

            [Bindable]
            private var message:String;

            [Bindable]
            private var formIsValid:Boolean = false;

            [Bindable]
            public var formIsEmpty:Boolean;

            // Holds a reference to the currently focussed control on
the form.
            private var focussedFormControl:DisplayObject;

            private function handleSaveResult(ev:ResultEvent):void {
                clearFormHandler();
                validateForm(ev);
                Alert.show("Código criado/atualizado com sucesso.",
"Informação", Alert.OK, null, null, null, Alert.OK);
            }

            parentApplication.listarCodigos.loaderService.getCodigos();
            PopUpManager.removePopUp(this);
        }

        private function handleFault(ev:FaultEvent):void {
            message = "Error: " + ev.fault.faultCode + " \n "
                + "Detail: " + ev.fault.faultDetail + " \n "
                + "Message: " + ev.fault.faultString;
        }

        public function saveCode():void {
            novoCodigo.addUpdateCodigo(codigo);
        }
    </mx:Script>

```

```

        private function creationCompleteHandler():void {
            PopUpManager.centerPopUp(this);
            resetFocus();
        }

        private function resetFocus():void {
            focusManager.setFocus(nomeTextInput);
        }

        // Validate the form
        public function validateForm(event:Event):void {
            focussedFormControl = event.target as
DisplayObject;

            formIsValid = true;

            formIsEmpty = (nomeTextInput.text == "");

            validate(nomeValidator);
            validate(comentarioValidator);

        }

        private function validate(validator:Validator):Boolean {
            // See also here for validating data:
            //
http://www.adobe.com/devnet/flex/quickstart/validating\_data/

            var validatorSource:DisplayObject =
validator.source as DisplayObject;
            var suppressEvents:Boolean = (validatorSource !=
focussedFormControl);
            var event:ValidationResultEvent =
validator.validate(null, suppressEvents);
            var currentControlIsValid:Boolean = (event.type ==
ValidationResultEvent.VALID);

            formIsValid = formIsValid && currentControlIsValid;
            return currentControlIsValid;
        }

        private function clearFormHandler():void {
            // Clear all input fields.
            nomeTextInput.text = "";
            comentarioTextInput.text = "";

            message = "";

            // Clear validation error messages.
            nomeTextInput.errorString = "";
            comentarioTextInput.errorString = "";

            formIsEmpty = true;
            formIsValid = false;
            resetFocus();
        }
    ]]>
</mx:Script>

<mx:StringValidator id="nomeValidator" source="{nomeTextInput}"
property="text" minLength="2" required="true" />

```

```

    <mx:StringValidator id="comentarioValidator"
source="{comentarioTextInput}"      property="text" minLength="2"
required="false" />

    <mx:Form width="100%" height="162" >
        <mx:FormItem label="Nome" required="true">
            <mx:TextInput id="nomeTextInput"      text="{codigo.nome}"
change="validateForm(event);" width="345"/>
        </mx:FormItem>
        <mx:FormItem label="Comentário" required="false">
            <mx:TextInput id="comentarioTextInput"
text="{codigo.comentario}"      change="validateForm(event);"
width="345" height="98"/>
        </mx:FormItem>
    </mx:Form>

    <mx:Spacer height="10" y="121" x="19"/>
    <mx:Text text="{message}" fontWeight="bold" width="300" x="10"
y="170"/>
    <mx:ControlBar height="43" y="225" horizontalAlign="center">
        <mx:Button label="Salvar código" id="submitButton"
enabled="{formIsValid}"      click="saveCode();" />
        <mx:Button label="Cancelar"

click="PopUpManager.removePopUp(this);" />
    </mx:ControlBar>

</mx:TitleWindow>

```

O arquivo ListarCodigos.mxml ilustra a tela da aplicação responsável por realizar a listagens de códigos existentes, fornecendo opções para criação/atualização/remoção de códigos.

Arquivo ListarCodigos.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Panel
xmlns:mx="http://www.adobe.com/2006/mxml"
xmlns:view="br.ufpb.di.webqda.view.*"
layout="absolute"
borderColor="#9AC2FF"
title="Códigos - Encontrados {codigoRecords} registros."
creationComplete="loadCodigos();"
height="70%"
width="70%">

    <mx:RemoteObject id="loaderService" destination="codigoService"
result="handleLoadResult(event)"      fault="handleFault(event)"
showBusyCursor="true" />
    <mx:RemoteObject id="deleteService" destination="codigoService"
result="handleDeleteResult(event)"      fault="handleFault(event)"
showBusyCursor="true" />

    <mx:Script>
        <![CDATA[

```

```

import br.ufpb.di.webqda.vo.Codigo;
import mx.controls.Alert;
import mx.managers.PopUpManager;
import mx.containers.TitleWindow;
import mx.collections.ArrayCollection;
import mx.rpc.events.ResultEvent;
import mx.rpc.events.FaultEvent;

[Bindable]
private var message:String;

[Bindable]
private var codigos:ArrayCollection = new
ArrayCollection();

[Bindable]
private var codigoRecords:int = 0;

public function loadCodigos():void {
    loaderService.getCodigos();
}

private function deleteCodigo():void {
    if(dataGrid.selectedItem != null) {
        var selectedItem:Codigo =
dataGrid.selectedItem as Codigo;
        deleteService.deleteCodigo(selectedItem.id);
    }
}

private function createCodigo():void {
    var titleWindow:NovoCodigo =
NovoCodigo(PopUpManager.createPopUp(this, NovoCodigo, true));
    titleWindow.setStyle("borderAlpha", 0.9);
    titleWindow.formIsEmpty = true;
}

private function updateCodigo():void {
    var titleWindow:NovoCodigo =
NovoCodigo(PopUpManager.createPopUp(this, NovoCodigo, true));
    titleWindow.setStyle("borderAlpha", 0.9);
    titleWindow.codigo = dataGrid.selectedItem as
Codigo;
    titleWindow.formIsEmpty = false;
}

private function handleLoadResult(ev:ResultEvent):void {
    codigos = ev.result as ArrayCollection;
    codigoRecords = codigos.length;
}

private function handleDeleteResult(ev:ResultEvent):void
{
    Alert.show("Código excluído com sucesso.",
"Informação", Alert.OK, null, null, null, Alert.OK);
    loadCodigos();
}

private function handleFault(ev:FaultEvent):void {
    message = "Error: "
+ ev.fault.faultCode + " - "

```

```

        + ev.fault.faultDetail + " - "
        + ev.fault.faultString;
    }

    private function fecharPanel():void {
        this.visible=false;
        parentApplication.login.appMenuBar.enabled=true;
    }
    ]]>
</mx:Script>

<mx:VBox width="100%" height="100%">
    <mx:Label text="{message}" fontWeight="bold"
includeInLayout="false" />
    <mx:DataGrid
        id="dataGrid"
        width="100%"
        height="100%"
        alternatingItemColors="[#F7F7F7, #E3F0FF]"
        dataProvider="{codigos}"
        doubleClickEnabled="true"
        doubleClick="updateCodigo()" >
        <mx:columns>
            <mx:DataGridColumn dataField="id"
headerText="ID Código" width="100"/>
            <mx:DataGridColumn dataField="nome"
headerText="Nome" />
            <mx:DataGridColumn dataField="comentario"
headerText="Comentário" />
            <mx:DataGridColumn dataField="idfamilia"
headerText="ID Família" />
            <mx:DataGridColumn dataField="data"
headerText="Criado" />
            <!-- <mx:DataGridColumn dataField="data_atualizacao"
headerText="Modificado" />-->
        </mx:columns>
    </mx:DataGrid>
</mx:VBox>
<mx:ControlBar horizontalAlign="center">
    <mx:Button label="Criar código" click="createCodigo()"
tooltip="Cria um novo código e o armazena na base de dados." />
    <mx:Button label="Atualizar código" click="updateCodigo()"
enabled="{dataGrid.selectedItem}" tooltip="Atualiza um código
existente na base de dados." />
    <mx:Button label="Excluir código" click="deleteCodigo()"
enabled="{dataGrid.selectedItem}" tooltip="Exclui o código da
base de dados." />
    <mx:Button label="Recarregar dados" click="loadCodigos()"
tooltip="Recarrega a lista de códigos da base de dados." />
    <mx:Button label="Fechar"
click="fecharPanel()" tooltip="Fecha este painel."/>
</mx:ControlBar>

</mx:Panel>

```

APÊNDICE D – TELAS DO WEBQDA

São apresentadas a seguir telas da ferramenta proposta, ilustrando aspectos da camada de apresentação, bem como algumas funcionalidades do WebQDA.

webqda Análise qualitativa de dados na Web de forma colaborativa

Sobre o WebQDA

O WebQDA (QDA - acrônimo para Qualitative Data Analysis) é uma aplicação que visa auxiliar a análise qualitativa de dados. A mesma foi projetada para contemplar o conceito de colaboração, permitindo um aumento da produtividade na pesquisa qualitativa através do uso da Web.

-  Acesse seus projetos de qualquer lugar
-  Comunique-se com os colaboradores dos seus projetos
-  Aproveite as vantagens das redes sociais

Já possui uma conta?

Login:

Senha:

Entrar

Novo no WebQDA?

Ainda não possui uma conta?

Cadastre-se agora!

Cadastrar

WebQDA © 2010

Figura 13 - Tela de login do WebQDA.

A Figura 13 apresenta a tela inicial do WebQDA, através da qual os pesquisadores envolvidos com análise qualitativa de dados poderão acessar a aplicação.

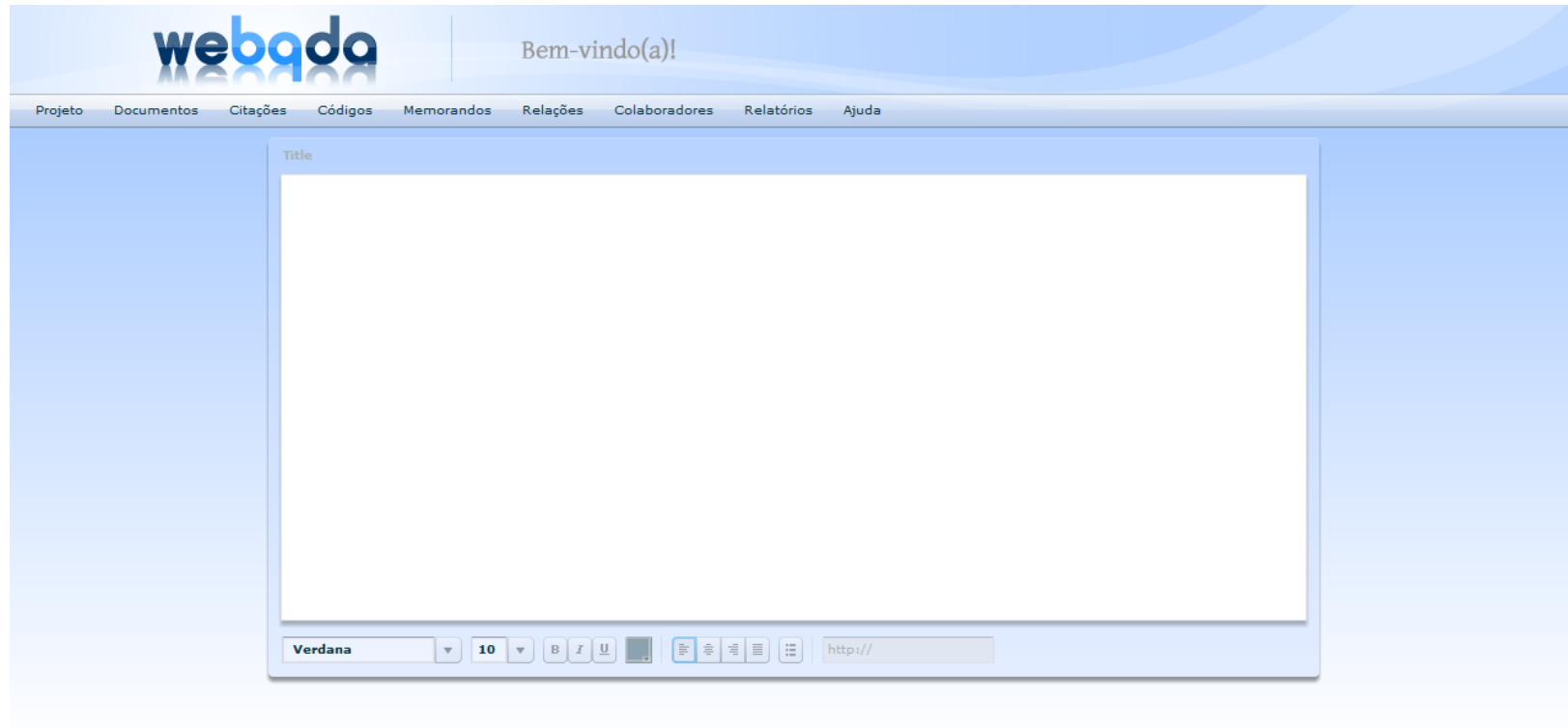


Figura 14 - Workspace do WebQDA.

A Figura 14 apresenta a *workspace* do WebQDA, através da qual os pesquisadores poderão criar seus projetos, criar os códigos que serão utilizados nas suas pesquisas, bem como executar as demais operações que podem ser realizadas através da barra de menu presente na aplicação.

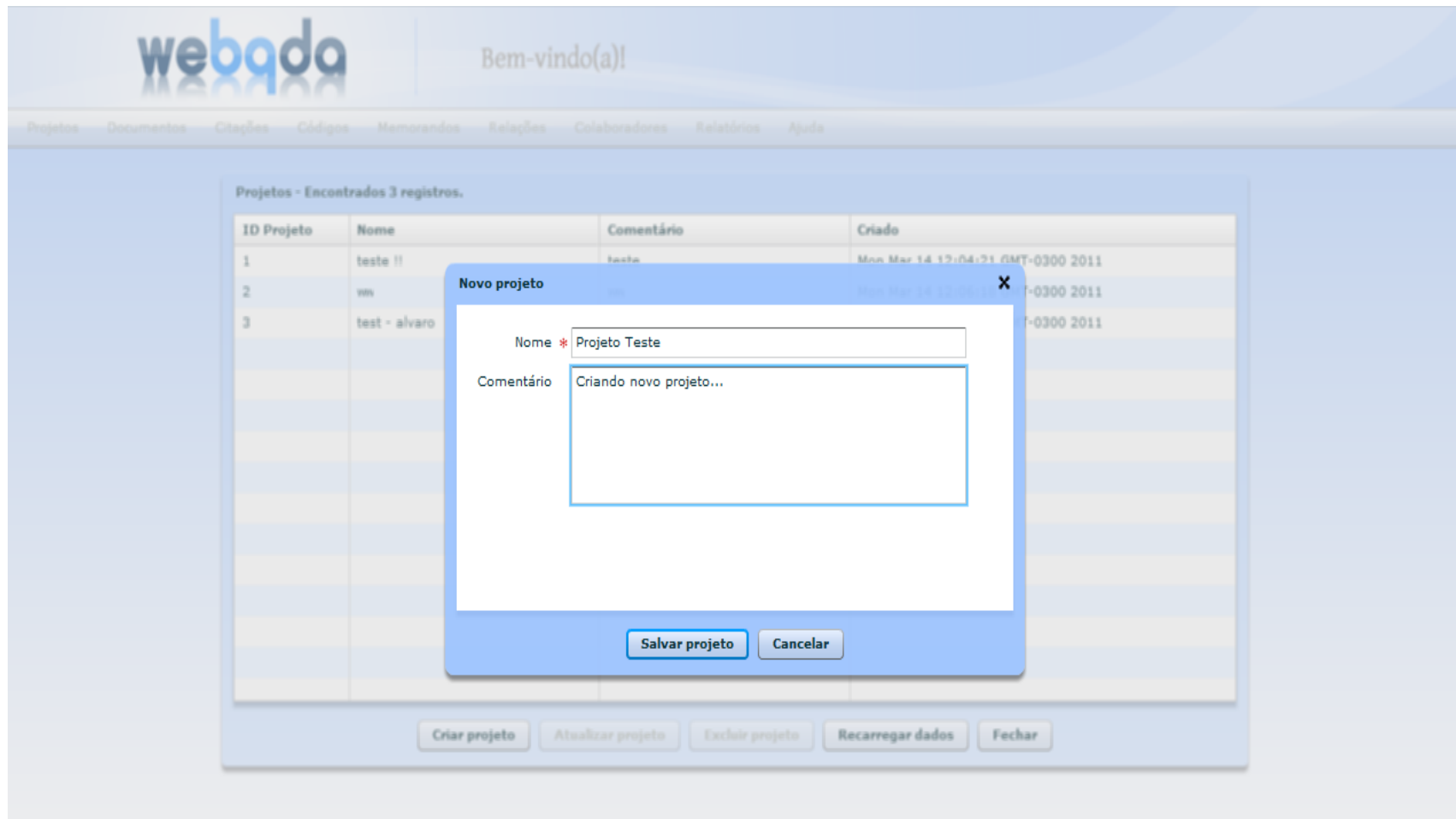


Figura 15 - Novo projeto.

A Figura 15 ilustra a criação de um novo projeto, sendo passados pelo usuário um nome e um comentário (opcional) para o mesmo.

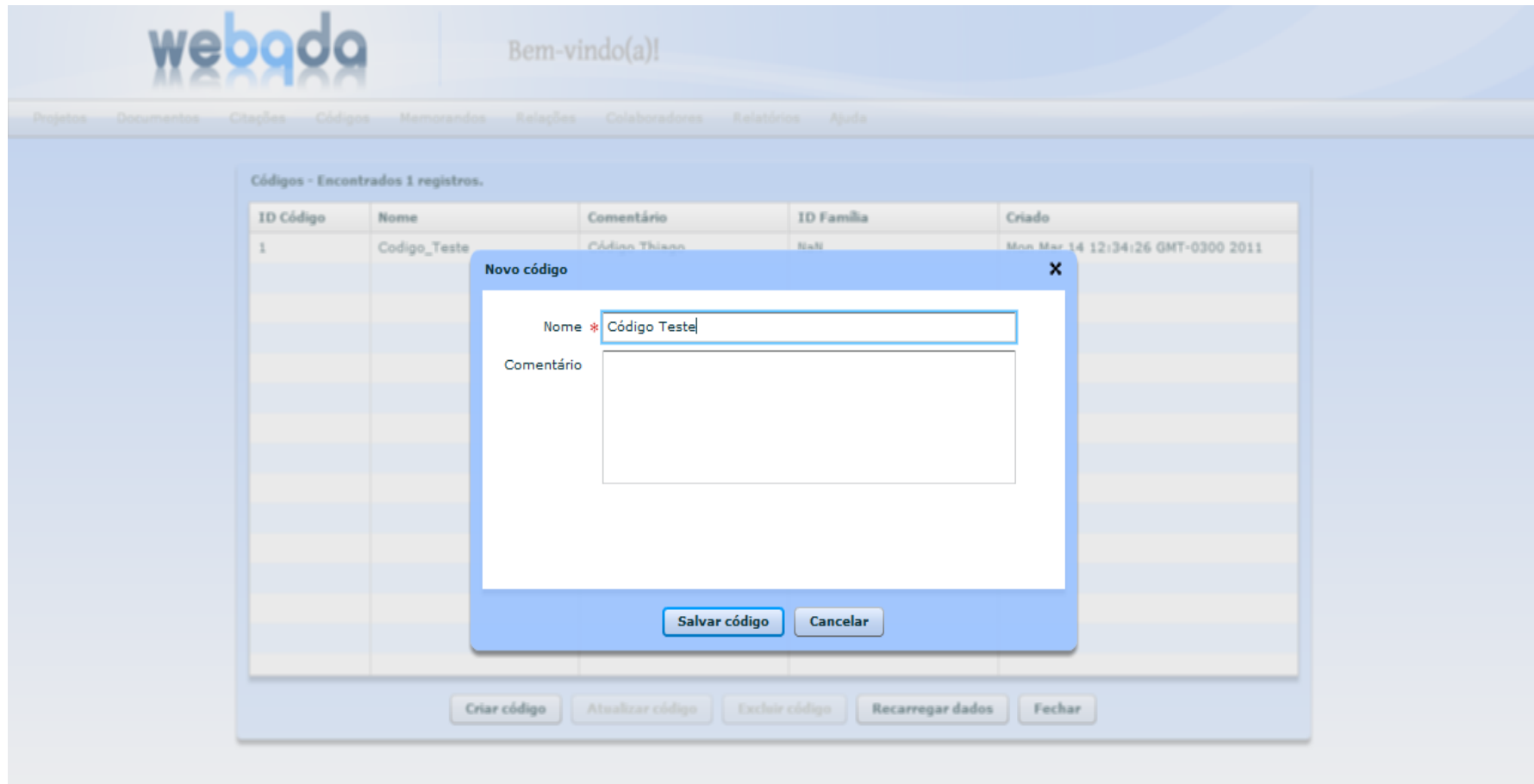


Figura 17 – Novo código.

A Figura 17 ilustra a criação de um novo código, sendo passados pelo usuário um nome e um comentário (opcional) que pode ser usado para explicar o significado do mesmo.

