

Modelo de uma Biblioteca Virtual com Peer-to-Peer e Web Services

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística

Fábio Schmitz Tani

Florianópolis, 2004

Resumo

Apesar do grande poder de conectividade de aplicações distribuídas, em especial redes de troca de arquivos *peer-to-peer*, o conteúdo dessas redes públicas tende a ser duvidoso em suas fontes, e muitas vezes ilegal.

Usando um esquema de indexação centralizada, e controle de publicação de conteúdo, torna-se possível o controle do conteúdo dessas redes, possibilitando que o conteúdo da rede seja confiável, e assim tornando esse estilo de rede *peer-to-peer* em um ambiente propício para a criação de aplicações que precisam garantir que o conteúdo da rede seja apenas o permitido, no caso deste trabalho, uma biblioteca virtual.

Palavras-chave: P2P, *peer-to-peer*, compartilhamento de arquivos, metadados, redes distribuídas.

Abstract

In spite of the great connectivity power of distributed computing, in special peer-to-peer file-sharing networks, the content of these public networks is not reliable and most times illegal in its form.

Using centralized indexing and controlling the publication of content, the control of the content of these networks is possible, and in doing so, allowing the content of these networks to be reliable, that way this form of peer-to-peer network can become a suitable environment for applications that need content control, in the case of this paper, a virtual library.

Key-words: P2P, *peer-to-peer*, file sharing, metadata, distributed networks.

Sumário

Sumário	4
1. Introdução	8
1.1 Importância do tema.....	8
1.2 Objetivo	8
1.2.1 Geral.....	8
1.2.2 Específico	9
1.3 Justificativa	9
1.3.1 Social.....	9
1.3.2 Científica	9
1.3.3 Pessoal.....	10
1.4 Metodologia de trabalho.....	10
1.4.1 Método científico	10
1.4.2 Pesquisa.....	11
2. Fundamentação Teórica.....	12
2.1 <i>PEER-TO-PEER</i>	12
2.1.1 Diferenças entre ponto-a-ponto e <i>peer-to-peer</i>	13
2.1.2 Tipos de aplicações <i>peer-to-peer</i>	14
2.1.3 Elementos de uma rede <i>peer-to-peer</i>	15
2.1.3.1 <i>Peers</i>	15
2.1.3.2 Grupos de <i>peers</i>	16
2.1.3.3 Transporte de rede.....	17
2.1.3.4 Serviços.....	17
2.1.3.5 Propaganda.....	18
2.1.3.6 Protocolos	18
2.1.3.7 Nome de entidades	18
2.1.4 Projeto JXTA	19
2.2 Web services	19
2.2.1 SOAP.....	21
2.3 Metadados.....	21
2.3.1 Resource Description Framework.....	22
2.3.1.1 Modelo RDF	22
2.3.1.2 Sintaxe do RDF/XML	23
2.4 Segurança e criptografia	24
2.4.1 Hash	25
2.4.2 Criptografia.....	26
2.4.2.1 Chave pública	26
2.5 UML	27
2.5.1 Casos de uso	27
2.5.1.1 Atores.....	28
2.5.2 Diagramas de seqüência	28
2.5.3 Modelo.....	29
2.5.4 Diagramas de casos de uso.....	29
3. Proposta DE um modelo de biblioteca virtual.....	30
3.1 Segurança	30
3.1.1 Segurança no JXTA	31
3.1.2 Segurança para a biblioteca virtual.....	32
3.1.2.1 Nodos autoridades	32
3.1.2.2 Nodos credenciadores	34

3.2	QoS.....	35
3.2.1	QoS em um ambiente <i>peer-to-peer</i>	35
3.2.2	QoS aplicado à biblioteca virtual.....	36
3.2.2.1	Balanceamento de carga (Rede).....	36
3.2.2.2	Repositórios dinâmicos de informação.....	37
3.2.2.3	Redundância e tolerância à falhas	38
3.2.2.4	Balanceamento de carga (Cliente)	38
4.	Análise DO MODELO DE BIBLIOTECA VIRTUAL para uma biblioteca virtual pública.....	39
4.1	Visão dos atores.....	39
4.2	Funcionamento da rede	40
4.2.1	Indexação centralizada	40
4.2.2	Segurança de conteúdo	41
4.3	Funcionalidades básicas	42
4.3.1	Pesquisa.....	42
4.3.2	Download	43
4.3.3	Compartilhamento.....	43
4.4	Funcionalidades específicas	43
4.5	Descrição do modelo proposto	43
4.5.1	Caso de uso expandido: Procura de Arquivos	44
	Contratos.....	45
4.5.2	Caso de uso expandido: Fazer Download de Arquivos.....	45
	Contratos.....	46
4.5.3	Caso de uso expandido: Publicar Arquivos	47
	Contratos.....	48
4.5.4	Caso de uso expandido: Retirar Arquivo Publicado	48
	Contratos.....	49
4.5.5	Caso de uso expandido: Credenciar Arquivo Publicado	50
	Contratos.....	52
4.5.6	Caso de uso expandido: Remover Credencial	53
	Contratos.....	54
4.6	Modelo Conceitual.....	54
5.	Conclusões e trabalhos futuros	56
5.1	Conclusão.....	56
5.2	Trabalhos futuros.....	56
6.	Referências Bibliográficas	58
7.	ANEXO A	61
1.	PEER-TO-PEER	61
1.1	Elementos de uma rede <i>peer-to-peer</i>	63
1.1.1	Peers	63
1.1.2	Grupos de <i>peers</i>	64
1.1.3	Transporte de rede.....	65
1.1.4	Serviços.....	65
1.1.5	Propaganda.....	66
1.1.6	Protocolos	66
1.1.7	Nome de entidades	66
1.1.8	Projeto JXTA	67
2.	modelo de biblioteca virtual	68
2.1	Segurança	68
2.2	Segurança para a biblioteca virtual.....	68

2.2.1	Nodos autoridades	69
2.2.2	Nodos credenciadores	71
3.	Análise DO MODELO DE BIBLIOTECA VIRTUAL para uma biblioteca virtual pública	72
3.1	Modelo Conceitual.....	72
4.	Referências Bibliográficas	74

Lista de Figuras

Figura 2.1a – Arquitetura Cliente/Servidor.....	12
Figura 2.1b – Arquitetura <i>peer-to-peer</i>	13
Figura 2.2a – Passos para acesso a um web service	20
Figura 2.2.1a – A estrutura da mensagem SOAP	21
Figura 2.3.1.1a – declaração RDF	22
Figura 2.3.1.1b – modelo RDF	23
Figura 2.3.1.2a – exemplo de sintaxe do RDF/XML	23
Figura 3.1.2.1a – Rede <i>peer-to-peer</i> com nodo autoridade	33
Figura 3.1.2.1b – Esquema de publicação	34
Figura 3.2.2.1a – Nível de conteúdo	37
Figura 4.1a – Diagrama de Casos de Uso.....	40
Figura 4.5.1a – Diagrama de seqüência do caso de uso da procura de arquivos	45
Figura 4.5.2a – Diagrama de seqüência do caso de: Fazer download de arquivos .	46
Figura 4.5.3a – Diagrama de seqüência do caso de uso: Publicar Arquivos	48
Figura 4.5.4a – Diagrama de seqüência do caso de uso: Retirar Arquivo Publicado	49
Figura 4.5.5a – Diagrama de seqüência principal do caso de uso: Credenciar Arquivo Publicado.....	51
Figura 4.5.5b – Diagrama de seqüência secundária do caso de uso: Credenciar Arquivo Publicado	52
Figura 4.5.6a – Diagrama de seqüência do caso de uso: Remover Credencial	54
Figura 4.6a – Modelo Conceitual da Biblioteca Virtual	55

1. INTRODUÇÃO

1.1 Importância do tema

Toda produção científica gerada, deve possuir pesquisas em materiais científicos, que sejam válidos. Muitas vezes, essas pesquisas tornam-se lentas devido ao formato em que essas informações se encontram, ou da falta de divulgação do material.

As produções científicas, como trabalhos de conclusão de curso, dissertações e teses, são de domínio público e encontram-se em seus respectivos departamentos da universidade, entretanto, é necessário, primeiramente, saber que tais produções existem antes que se possam procurar por elas.

Por meio de metadados podemos agregar maior informação a arquivos e assim possibilitar uma procura por características, tais como palavras-chave, tipo da produção, entre outros. Esse conjunto de metadados facilita a busca de arquivos nos repositórios de dados.

O mercado brasileiro de *peer-to-peer* (P2P) é aparentemente inexistente, soluções que se utilizam dessa arquitetura de programação, através de uma pesquisa simples pode-se localizar apenas a empresa PiX Insight, que parece usar da arquitetura *peer-to-peer* em suas aplicações [WEBSOL 2004].

1.2 Objetivo

1.2.1 Geral

Modelar um ambiente *peer-to-peer*, que disponibilize conteúdo científico para pesquisas da comunidade acadêmica a fim de apoiar o processo de geração de

conhecimentos científicos, fornecendo uma fonte de documentos virtuais confiáveis e seguros.

1.2.2 Específico

- a) Modelar um ambiente *peer-to-peer* que enfatize a pesquisa de documentos;
- b) Achar meios para garantir a autenticidade dos documentos em um ambiente *peer-to-peer*;
- c) Definir um conjunto de metadados para auxiliar no processo de pesquisa.

1.3 Justificativa

1.3.1 Social

A internet é uma indiscutível fonte de informação, entretanto não é possível considerá-la uma fonte confiável para pesquisas científicas, uma vez que vários textos são reproduzidos identicamente em diferenciados sites, sem dar devido crédito aos autores originais. A internet tem ilimitado potencial para proporcionar rapidez e dinamicidade para pesquisas, este, porém ainda não utilizado.

1.3.2 Científica

O desenvolvimento de aplicações *peer-to-peer* utiliza a mais recente abordagem sobre a internet para transferência de arquivos via rede, assim como balanceamento de carga.

O *peer-to-peer* é uma tecnologia recente que pode ser vista em ferramentas como os de troca de arquivos (*file sharing*) KaZaA, WinMX e o mais recente Bit

Torrent (*file swarming*¹). São aplicativos que sustentam a idéia de que, cada nodo deve compartilhar com a rede *peer-to-peer* seus dados e largura de banda, a fim de tornar mais eficiente a troca de arquivos. Quanto mais fontes, mais rápido os arquivos se espalham.

No caso dos *web services*, como o aplicativo é destinado a fins de pesquisa acadêmica, sua utilidade é de possibilitar que o cliente possa, por intermédio de seu browser, conectar-se ao servidor e utilizar o serviço para fazer buscas de arquivos sem necessitar instalar a ferramenta de procura. Desta forma facilita o acesso aos dados.

1.3.3 Pessoal

Como acadêmico, desejo contribuir com a comunidade, possibilitando a maior divulgação dos materiais gerados pela própria comunidade acadêmica e, utilizar de um meio já existente para facilitar o processo de pesquisa de material de caráter científico, preservando os direitos e a devida menção aos verdadeiros autores das produções.

1.4 Metodologia de trabalho

1.4.1 Método científico

Segundo Cervo e Berviam (1996), e também citado na obra de Falshin (1993) e Salomon (1978), o método científico é basicamente um método de investigação que consiste em um conjunto de processos. Estes processos foram utilizados por vários pesquisadores através dos tempos e reunidos de forma a criar um procedimento sistemático e ordenado para pesquisa científica.

¹ Os arquivos são divididos ao nível dos bits, não é necessário ter um arquivo completo para dividir o seu conteúdo.

1.4.2 Pesquisa

A pesquisa definida como:

“Procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. (...) A pesquisa é desenvolvida mediante o concurso dos conhecimentos disponíveis e a utilização cuidadosa de métodos, técnicas e outros procedimentos científicos (...) ao longo de um processo que envolve inúmeras fases, desde a adequada formulação do problema até a satisfatória apresentação dos resultados” [GIL 1996].

2. FUNDAMENTAÇÃO TEÓRICA

2.1 PEER-TO-PEER

O termo *peer-to-peer* apesar de parecer um termo novo, é na verdade baseado na primeira forma de implementação das redes, onde elas eram ligadas ponto a ponto, um nodo ao outro. [ORAM 2001]

Com o crescimento da internet, o ponto-a-ponto foi dando espaço à arquitetura cliente/servidor (Figura 2.1a), que é a atual estrutura da internet, onde vários provedores de serviços servem vários clientes. Na arquitetura cliente/servidor, os clientes apenas usam os serviços, e nunca colaboram com a rede, muitas vezes por causa de *gateways*² e/ou *firewalls*³, que tiram a habilidade deles de se tornarem provedores de serviços.

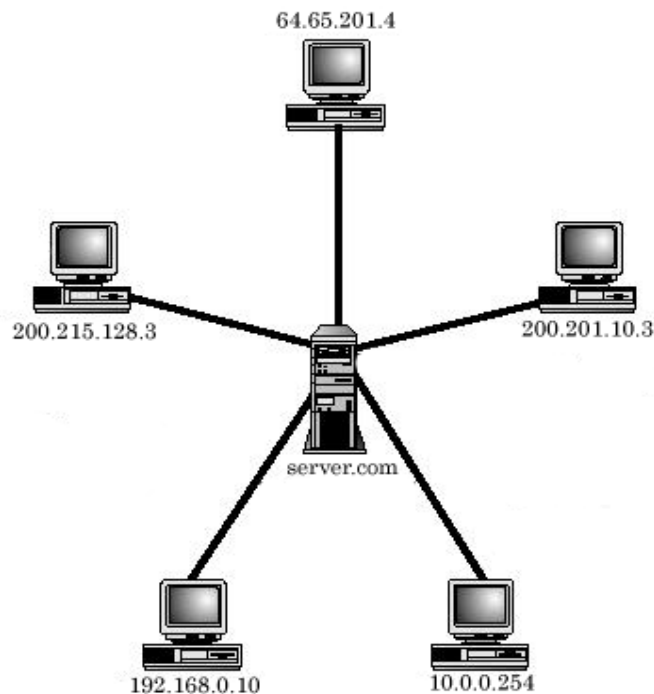


Figura 2.1a – Arquitetura Cliente/Servidor

² Nodo da rede que liga uma rede privada com outra rede. Ex.: Gateway de internet liga os nodos da rede local à internet.

³ Nodo da rede ou software que bloqueia tráfego para dentro ou para fora da rede, usado para proteger os recursos da rede contra invasores.

A recente arquitetura *peer-to-peer*, nada mais é do que a re-implementação da velha rede ponto-a-ponto adequando-a as redes de hoje em dia, possibilitando aos nodos tornarem-se provedores de serviços, mesmo atrás de *gateways* e *firewalls* (Figura 2.1b).

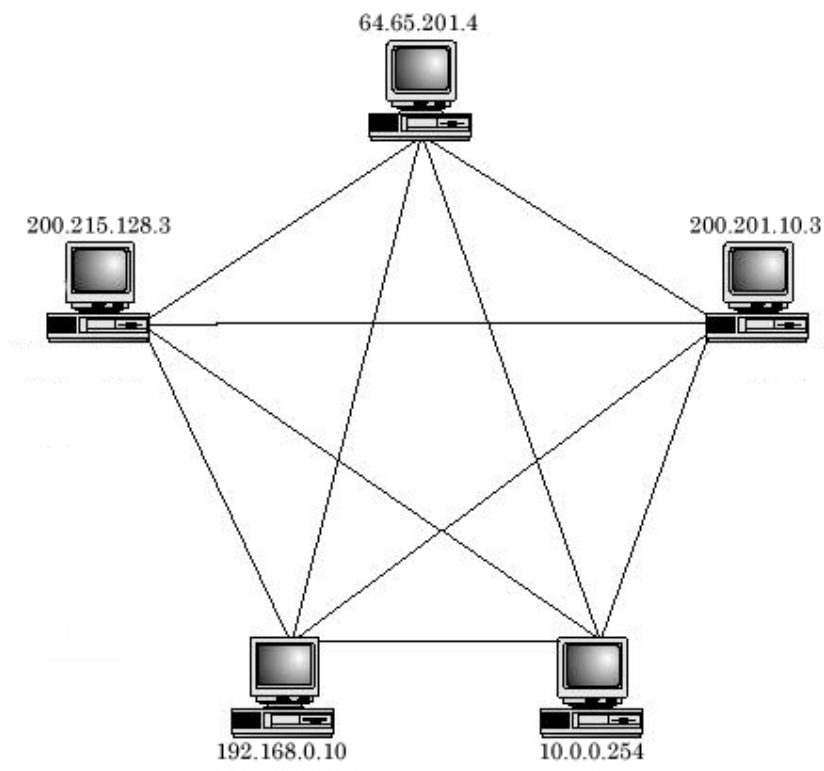


Figura 2.1b – Arquitetura *peer-to-peer*

2.1.1 Diferenças entre ponto-a-ponto e *peer-to-peer*

As duas arquiteturas, embora possuam os mesmos fundamentos, funcionam completamente diferente. O ponto-a-ponto consiste de uma rede TCP/IP onde cada nodo é conectado diretamente a cada outro nodo, por isso ponto-a-ponto.

O *peer-to-peer*, por causa dos *gateways* e *firewalls*, precisa usar de tecnologias para possibilitar que esses nodos possam se juntar a rede.

As redes cliente/servidor, usam-se do DNS, para mascarar os IPs numéricos, em endereços URL que facilitam a memorização, entretanto, o DNS foi projetado

para redes “no qual uma alteração no endereço IP era considerada anormal, rara e podia levar dias para se propagar pelo sistema” [ORAM 2001]. O DNS não serve ao propósito do *peer-to-peer*, pois os nodos estão constantemente se conectando e desconectando na rede.

As aplicações de hoje em dia, utilizam de outros métodos alternativos ao DNS para localizar os nodos da rede. Um desses métodos é a criação de um repositório de endereços IP próprios, que funcionam em tempo real e utilizado, por exemplo, pelo Napster e ICQ. Um segundo método é ignorar o DNS e utilizar diretamente do IP numérico, como é feito com o Gnutella⁴ e o Freenet⁵. Outro é deixar os clientes se conectar periodicamente a um endereço fixo, aonde trocam seus IPs e atualizam seus endereços, como o SETI@home⁶ faz [ORAM 2001].

2.1.2 Tipos de aplicações *peer-to-peer*

Um dos mais conhecidos estilos de aplicativos *peer-to-peer*, foram os programas de troca de arquivo, a exemplo de Napster e KaZaA, essas aplicações ajudaram ao grande crescimento das redes *peer-to-peer*. Outro estilo de *peer-to-peer* muito usado hoje em dia, são os *Instant Messengers*⁷, como o ICQ, que foi o primeiro aplicativo a fazer com que usuários interagissem entre si diretamente, ao invés de se conectar a um prestador de serviços.

Podemos classificar os estilos de aplicações *peer-to-peer* elaboradas até hoje em dia:

⁴ O Gnutella é uma ferramenta P2P, sobre a licença GNU, para compartilhamento de arquivos.

⁵ O Freenet é um sistema descentralizado de distribuição de arquivos, que visa evitar a censura de documentos, proporcionar anonimato aos usuários, armazenar e distribuir documentos eficientemente.

⁶ O SETI tem a função de buscar inteligência extraterrestre (Search for Extraterrestrial Intelligence), o SETI@HOME é a implementação P2P que busca dividir a tarefa de analisar os dados encontrados com a antena.

⁷ Programas de troca de mensagens instantâneas

- *Instant Messengers*: Visam proporcionar troca de mensagens instantaneamente, usando de apelidos, e-mails, ou números identificadores para diferenciar um nodo do outro. A comunicação é feita diretamente entre os nodos, uma vez que os dois se identificaram no servidor de nomes;
- Troca de arquivos: Uma das formas mais conhecidas de *peer-to-peer*, tornado famoso por aplicações a estilo do Napster e KaZaA. Usando algumas vezes de um servidor central, outras vezes de uma forma distribuída, utilizam a rede *peer-to-peer*, para fazer trocas de quaisquer tipos de arquivos, desde músicas a livros e software;
- Computação Distribuída: É usada uma rede *peer-to-peer* com nodos interconectados que fazem pequenas tarefas, e contribuem de alguma forma para um objetivo central, gerenciado por um ou vários servidores. Um exemplo é o projeto SETI@home, que busca por vida extraterrestre, analisando sinais de satélite. Cada nodo executa uma pequena parte da tarefa, e periodicamente se comunica com um dos servidores para pegar uma nova tarefa.

2.1.3 Elementos de uma rede *peer-to-peer*

2.1.3.1 Peers

Um *peer* é um nodo em uma rede *peer-to-peer*, e é definido como “qualquer entidade capaz de fazer algum trabalho útil e comunicar os resultados desse trabalho para outra entidade dentro da rede, direta ou indiretamente” [WILSON 2002].

Essa definição de trabalho útil depende do tipo do *peer*, os quais podem ser separados em três tipos:

- **Peers simples:** São usuários simples, normalmente atrás de *firewalls*, por causa do seu acesso limitado eles tem o menor nível de responsabilidade na rede;
- **Peers rendez-vous:** Responsáveis por proporcionar aos outros *peers* da rede informações sobre outros *peers* que eles têm conhecimento. Essas informações são obtidas através de uma *discovery query* feita por um *peer* ao rendez-vous peer;
- **Peers Roteadores:** Fazem a ponte entre os *peers* que estão atrás de *firewalls* ou *gateways*. Agem de forma similar aos DNS, mas de forma dinâmica, e providenciam os *peers* atrás da rede com uma representação que pode ser usada para se comunicar com *peers* fora da rede.

2.1.3.2 Grupos de *peers*

As aplicações de hoje em dia, pela sua natureza proprietária e especializada, acabaram dividindo o espaço da rede *peer-to-peer* entre as redes de cada aplicação, a exemplo, as redes Gnutella que se comunicam somente com redes Gnutella. Quando a rede *peer-to-peer* é composta de aplicações que usam os mesmos protocolos, como é feito com o JXTA, é necessário o conceito de grupos de *peers* para subdividir o espaço na rede.

Grupos de *peers* não são necessariamente abertos a qualquer nodo da rede. Eles podem ser divididos com base na aplicação utilizada, ou talvez necessitem de um nível de acesso de segurança por parte do *peer*. Alguns grupos, entretanto, requerem que o *peer* forneça algum serviço para poder se conectar ao grupo, ou aproveitar dos recursos do grupo.

2.1.3.3 Transporte de rede

Para efetuar a troca dos dados, os *peers* usam de algum mecanismo para troca de dados, seja um tipo de protocolo de transporte de baixo nível, a estilo do TCP ou UDP, ou de alto nível, como o HTTP ou o SMTP.

O conceito do transporte de rede pode ser quebrado em três partes, os *endpoints*⁸, os *pipes*⁹ e as *messages*¹⁰. Para enviar os dados de um *peer* ao outro, o *peer* remetente deve empacotar os dados a serem transmitidos em uma mensagem usando um *pipe* de saída, ligado ao *endpoint* que age como fonte dos dados, no outro extremo outro *peer* recebe a mensagem através de um *pipe* de entrada, ligado ao *endpoint* que age como destinação final, e extrai os dados transmitidos.

2.1.3.4 Serviços

São as funcionalidades que os *peers* na rede disponibilizam ao outros *peers*, e englobam todas as atividades que a rede de *peer-to-peer* foi projetada para fazer ou disponibilizar. Esses serviços podem ser divididos em duas categorias:

- **Serviços de *peers*:** Uma funcionalidade proporcionada por um dos nodos da rede, disponível somente quando este nodo estiver conectado a rede;
- **Serviços de grupos de *peer*:** Uma funcionalidade oferecida aos membros de um grupo, ela é disponibilizada por vários membros do grupo, provendo acesso redundante ao serviço. Enquanto um nodo estiver conectado, o serviço está disponível na rede.

⁸ “A fonte inicial ou destinação final de qualquer pedaço de dado sendo transmitida na rede” [WILSON 2002]

⁹ “Canais de comunicação unidirecionais, assíncronos e virtuais conectando dois ou mais *endpoints*” [WILSON 2002].

¹⁰ “Container para os dados sendo transmitidos sobre um *pipe* de um *endpoint* ao outro” [WILSON 2002]

2.1.3.5 Propaganda

“Uma representação estruturada de uma entidade, serviço, ou recurso tornado disponível por um *peer* ou grupo de *peers* como parte da rede *peer-to-peer*” [WILSON 2002].

2.1.3.6 Protocolos

Um protocolo é apenas uma forma de dizer como alguma ação deve ser feita, no caso do *peer-to-peer* eles são necessários para definir os tipos de interação que um *peer* pode fazer como parte da rede.

O uso da propaganda simplifica os protocolos necessários em uma rede *peer-to-peer*, pois elas definem a estrutura e representação dos dados. Os protocolos organizam então a troca de propagandas que contém a informação necessária. [WILSON 2002]

2.1.3.7 Nome de entidades

A maioria dos componentes de uma rede *peer-to-peer* precisa de um identificador único que as identifique. A rede *peer-to-peer* ideal qualquer dispositivo deve poder participar, independente de sistema operacional ou transporte de rede. [WILSON 2002]

Alguns exemplos de identificação necessários na rede *peer-to-peer*:

- **Peers:** Precisam ser identificados por uma entidade que permita que outros *peers* o localizem na rede;
- **Grupos de peers:** Um *peer* precisa saber qual o grupo em que ela fará alguma ação;

- **Pipes:** A comunicação necessita que seja identificado, qual *pipe* está conectado a qual *endpoint*;
- **Conteúdos:** O conteúdo da rede deve possuir um identificador único, que possibilite que os *peers* o identifiquem pela rede, e que haja redundância da informação entre os *peers*.

2.1.4 Projeto JXTA

É um *framework*¹¹ para aplicações *peer-to-peer*, foi iniciado como um projeto da Sun Microsystems, com o objetivo de explorar a visão da computação distribuída usando uma topologia *peer-to-peer*. O projeto continua em andamento, mas seu código é aberto, e conta com a ajuda de muitos membros da comunidade Java. Ele fornece toda a infra-estrutura para criar redes *peer-to-peer*, usando nodos, super-nodos¹², grupos e outras tecnologias do *peer-to-peer*.

Por ser um projeto em andamento, a cada dia surgem novas mudanças e alterações ao seu código, entretanto o desenvolvimento dessa ferramenta já está em um estado estável, que possibilita a criação de plataformas inteiras em cima do *framework*.

2.2 Web services

Um *web service* é um tipo de interface acessível pela rede para uma determinada funcionalidade de aplicação, e é construído utilizando tecnologias e

¹¹ Wirfs-Brock (apud Silva, 2000) “Um esqueleto de implementação de uma aplicação de um subsistema de aplicação, em um domínio de problema particular. É composto de classes abstratas e concretas e provê um modelo de interação ou colaboração entre as instâncias de classes definidas pelo framework. Um framework é utilizado através de configuração ou conexão de classes concretas e derivação de novas classes concretas a partir das classes abstratas do framework”.

¹² Nodos da rede que atuam como servidores com a finalidade de localizar os clientes conectados a rede P2P.

protocolos padrões da internet, como HTTP¹³ (*Hyper Text Transport Protocol*), XML¹⁴ (*eXtensible Markup Language*), SMTP¹⁵ (*Simple Mail Transport Protocol*), SOAP¹⁶ (*Simple Object Access Protocol*) entre outros.

Segundo ZDNet (2002) os passos para o acesso de um web service são dados da seguinte forma:

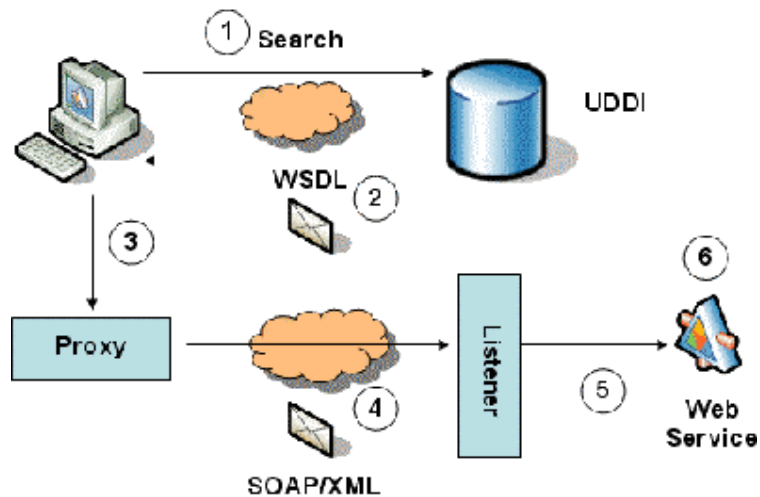


Figura 2.2a – Pasos para acesso a um web service

Se uma aplicação pode ser acessada via rede utilizando um dos protocolos acima, então ela é um *web service*.

O *web service* usa de uma interface WSDL (*Web Services Description Language*), que mostra ao cliente como as funções devem ser utilizadas, por sua vez o cliente implementa a interface do WSDL, e através dos stubs gerados pode acessar diretamente os serviços da aplicação.

¹³ Protocolo usado na internet para acessar páginas pelo browser, normalmente opera na porta 80 do computador.

¹⁴ Método padrão de identificação e descrição de dados na Web, é uma linguagem sintaticamente genérica, de fácil leitura para humanos e máquinas, de descrição de dados hierárquicos, aplicáveis a um conjunto enorme de aplicações, banco de dados, comércio eletrônico, Java, desenvolvimento web, busca e etc.

¹⁵ Protocolo usado para envio de mensagens eletrônicas (e-mail).

¹⁶ SOAP é baseado em XML e descreve um formato de mensagem para comunicação máquina-a-máquina.

2.2.1 SOAP

Para acesso a servidores atrás de *firewalls* ou NATs¹⁷ (*Network Address Translators*) é utilizado do protocolo SOAP, ele consiste de um envelope que engloba a mensagem, e seu cabeçalho possui informação suficiente para entregar o pacote ao nodo por trás do *firewall*.

Segundo Sun One (2004) a estrutura de uma mensagem SOAP pode ser representada da seguinte forma:

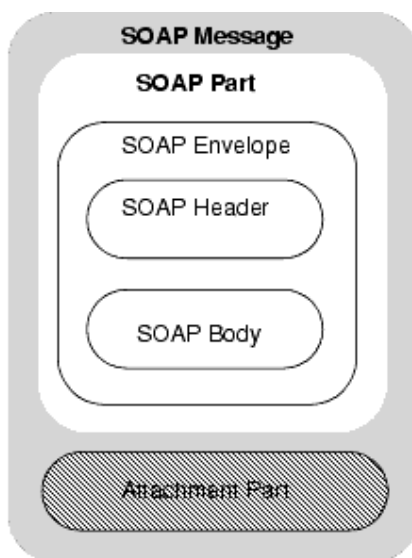


Figura 2.2.1a – A estrutura da mensagem SOAP

2.3 Metadados

Metadados são rótulos, como: autor, tipo, altura e idioma usados para descrever um livro, uma pessoa, um programa de televisão, a espécie, e etc. Um metadado é simplesmente um dado que descreve outro dado [ORAM 2001].

A Dublin Core Metadata Initiative (DCMI)¹⁸ é um instituição que criaram um conjunto de metadados, categorizados e catalogados, para vários meios, assim

¹⁷ Mascaram o IP de nodos atrás da rede local, para ser idênticos ao do gateway, dessa forma possibilitando que vários computadores acessem a internet utilizando apenas um IP válido, o do gateway.

¹⁸ Dublin Core Metadata Initiative – <http://www.dublincore.org>

diminuindo o trabalho da definição dos metadados para determinado documento, basta selecionar os metadados que se aplicam ao dado da lista de termos¹⁹ e adicionar a lista de metadados para o documento.

2.3.1 Resource Description Framework

O RDF (*Resource Description Framework*) é basicamente uma linguagem para a representação de recursos da Web. Foi projetado para disponibilizar as informações para aplicações, e não apenas exibir as informações para os usuários. O RDF então provê um *framework* comum entre aplicações para que haja troca de informações sem a perda de significado.

Usando de conjunto de metadados, como os da DCMI, é possível estruturá-los com o RDF, para possibilitar a troca de mensagens contendo informações sobre os dados em questão, no caso do foco do projeto, artigos, teses, e produções científicas no geral.

2.3.1.1 Modelo RDF

O RDF é baseado na idéia que as coisas sendo descritas possuem propriedades que tem valores, e que recursos podem ser descritos através de declarações (Figura 2.3.1.1a) que especificam essas propriedades e valores. [W3C 2004]

<p><code>http://www.example.org/index.html</code> tem uma data-de-criação a qual o valor é 16 de Agosto de 1999 <code>http://www.example.org/index.html</code> tem uma língua a qual o valor é Inglês</p>

Figura 2.3.1.1a – declaração RDF²⁰

A parte que identifica a coisa que a declaração fala sobre, é chamada de *subject* (assunto), a parte que identifica a propriedade ou característica do *subject* é

¹⁹ Lista de Termos - <http://www.dublincore.org/documents/2003/03/04/dcmi-terms/>

²⁰ Figura adaptada de W3C 2004.

chamada de *predicate* (predicado) e a parte que identifica o valor dessa propriedade é chamado de *object* (objeto).



Figura 2.3.1.1b – modelo RDF²¹

Pelo fato do RDF ser feito para que máquinas entendam as declarações, é necessária uma forma mais genérica de identificação, com *Uniform Resource Identifier*²² (URI) como forma básica de identificação dos *subject*, *predicate* e *object* da declaração. [W3C 2004].

Na verdade é utilizado *URI references* (URIref), que nada mais é que um URI com um fragmento identificador no final. E usa de XML para deixar as declarações em um formato possível para ser processado por máquinas, esse formato é chamado de *RDF/XML*.

2.3.1.2 Sintaxe do RDF/XML

O RDF/XML comporta-se basicamente como um documento XML, por usar de sua estrutura. Através de um exemplo básico [W3C 2004] podemos explicar melhor a sintaxe RDF/XML, para melhor entender seu funcionamento e estruturação.

```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:exterms="http://www.example.org/terms/">
4.   <rdf:Description rdf:about="http://www.example.org/index.html">
5.     <exterms:creation-date>16 de Agosto de 1999</exterms:creation-
6.   </rdf:Description>
7. </rdf:RDF>
```

Figura 2.3.1.2a – exemplo de sintaxe do RDF/XML

²¹ Figura adaptada de W3C 2004

²² Um URI é uma linha de caracteres compacta para identificar um recurso físico ou abstrato. [URIS 1998].

A primeira linha do exemplo (Figura 2.3.1.1c) é a declaração do XML, que indica que o conteúdo é XML e qual sua versão.

A segunda linha inicia um elemento `rdf:RDF` (que acaba na linha 7), seguido de uma declaração de *namespace*²³ representado como um `xmlns`, e especifica que todas as tags nesse contexto, com o prefixo `rdf:`, são parte do *namespace* identificado pelo URIref `http://www.w3.org/1999/02/22-rdf-syntax-ns#`. Essa URIref em questão é utilizada para termos do vocabulário RDF.

A terceira linha identifica outra declaração de *namespace*, para o prefixo `exterms:`, identificado pelo URIref `http://www.example.org/terms/`, usado para identificar o vocabulário usado pela empresa exemplo, `example.org`.

Na quarta a sexta linha, temos a descrição da declaração, e o assunto é a página `http://www.example.org/index.html`, identificada na quarta linha pelo prefixo `rdf:about`. Na quinta linha podemos identificar uma propriedade do elemento, identificado por `exterms:creation-date`, aonde o prefixo representa o URIref descrito na terceira linha. Sendo esse o predicado, o valor 16 de Agosto de 1999 é o objeto. Por estar localizado dentro do elemento `rdf:Description`, isto indica que esta propriedade está relacionada ao recurso especificado pelo `rdf:about`. A sexta linha indica o término do elemento iniciado na quarta linha.

A sétima linha marca o final do elemento `rdf:RDF`, que encapsula o documento.

2.4 Segurança e criptografia

Um aspecto fundamental de softwares distribuídos é a questão da segurança, pois eles trabalham em ambientes não controlados, e precisam fornecer certo nível

²³ Uma coleção de nomes, identificada por uma URIref, usadas em documentos XML como tipos de elementos ou nomes de atributos. [W3C 1999]

de segurança, para garantir seu funcionamento adequado, assim como a veracidade de seus dados. A segurança pode ser dada de várias formas:

- Confiabilidade, a proteção de dados transmitidos de ataques passivos;
- Autenticação, para identificar se a comunicação é autêntica;
- Integridade, para garantir que as mensagens sejam recebidas como foram enviadas;
- Não repúdio, previne que tanto o remetente quanto o destinatário neguem o envio ou recebimento de uma mensagem;
- Controle de acesso, controlar ou limitar o acesso para serviços;
- Disponibilidade.

No caso de serviços distribuídos na internet a questão de segurança deve ser levada ainda mais a sério, pois os dados trafegam sobre uma rede pública, e dependendo da informação, é necessário que os mesmos sejam protegidos.

2.4.1 Hash

O hash é uma função de caminho único que dado uma mensagem m produz uma saída de tamanho fixo. “O propósito de uma função hash, é produzir uma impressão digital de um arquivo, mensagem, ou qualquer outro bloco de dados” [STALLINGS 1998]. É possível enumerar seis propriedades básicas que uma função de hash deve possuir:

- A função pode ser aplicada a um bloco de dados de qualquer tamanho;
- A função retorna uma saída de tamanho fixo;

- A função $H(x)$ é relativamente fácil de ser calculada para qualquer x , tornando as implementações tanto em software quanto em hardware, práticas;
- Para qualquer resultado h , é computavelmente impossível achar x tal que $H(x) = h$. Também referida como a propriedade de caminho único.
- Para qualquer bloco x , é computavelmente impossível achar um $y \neq x$ com $H(y) = H(x)$. Também referida como a baixa resistência a colisão (*weak collision resistance*);
- É computavelmente impossível achar qualquer par (x,y) tal que $H(x) = H(y)$. Também referida como a forte resistência a colisão (*strong collision resistance*).

Algumas funções de hash populares são o MD2, MD4, MD5 da RSA e o SHA-1 e SHA-2 da NIST.

2.4.2 Criptografia

“A arte de proteger a informação transformando-a em um formato ilegível, chamado texto cifrado. Somente os que possuem uma chave secreta podem decifrar a mensagem em um texto legível”[WEBOPIDIA 2004].

Podemos classificar os sistemas de criptografia em criptografia simétrica, ou assimétrica, também conhecida como criptografia de chave pública.

Uma chave para a criptografia é um conjunto de caracteres que através de um algoritmo de criptografia pode criptografar e descriptografar uma mensagem.

2.4.2.1 Chave pública

Na criptografia de chave pública, são gerados sempre duas chaves, uma pública e de conhecimento de todos, e uma privada (secreta) e de conhecimento

somente do destinatário da mensagem [WEBOPIDIA 2004]. As chaves funcionam de tal forma que uma chave consegue descriptografar uma mensagem criptografada com a outra, mas nunca uma chave consegue descriptografar uma mensagem criptografada com a mesma chave.

Com esse modelo de segurança, uma chave privada é gerada e deve ser secreta, outra chave pública é gerada e é usada por quaisquer pontos que quiserem contatar o ponto da chave privada. Assim é possível criptografar quaisquer mensagens com a chave pública, que o único ponto capaz de entender a mensagem será o que possuir a chave privada correspondente.

2.5 UML

“A *Unified Modeling Language* (UML) é uma linguagem para especificar, visualizar, construir e documentar artefatos de sistemas de software, assim como modelagem de negócios e outros sistemas. O UML representa uma coleção das melhores práticas de engenharia que se provaram bem sucedidas na modelagem de sistemas grandes e complexos” [UML 2003].

Abaixo são explicadas em mais detalhes as práticas utilizadas nesse projeto, para modelar sistema de uma biblioteca virtual.

2.5.1 Casos de uso

Um caso de uso é definido como um classificador representando uma unidade coerente de uma funcionalidade provida por um sistema, é representado por seqüências de mensagens trocadas entre o sistema e um ou mais atores [UML 2003].

Casos de uso são utilizados para poder descrever funções que o sistema poderá executar e quais os passos de interações podem acontecer ao executar essa funcionalidade.

Um caso de uso pode conter regras de negócio que descrevem certas regras a serem seguidas que são inerentes e específicas do negócio para o qual o sistema está sendo desenvolvido.

Casos de uso são representados por círculos com os nomes do caso de uso.

2.5.1.1 Atores

Um ator define um conjunto de papéis que os usuários de uma entidade podem exercer quando interagindo com a mesma. Um ator pode exercer um papel diferente, dependendo do caso de uso com o qual ele se comunica [UML 2003].

Atores são representados por bonecos de palito, com um nome que o identifica no sistema.

2.5.2 Diagramas de seqüência

Um diagrama de seqüência mostra uma interação, no qual um conjunto de mensagens é enviado entre atores e objetos do sistema, que mostra a chamada de procedimentos entre instâncias para efetuar uma desejada operação [UML 2003].

Um diagrama de seqüência possui duas dimensões, sendo que a vertical representa o tempo e a horizontal representa diferentes instâncias. Cada ator ou objeto no diagrama de seqüência podem ser ligados através de setas, representando mensagens ou estímulos.

2.5.3 Modelo

Um modelo captura a visão de um sistema concreto e, portanto, é uma abstração de um sistema concreto com certo propósito [UML 2003].

Um modelo contém todos os elementos necessários para representar um sistema completamente, de acordo com o seu propósito.

2.5.4 Diagramas de casos de uso

Diagramas de caso de uso mostram atores e casos de uso e seus relacionamentos. Através dos atores e seus relacionamentos com os casos de usos, podemos determinar as partes do sistema que cada ator pode ver, assim como generalizações entre os atores [UML 2003].

3. PROPOSTA DE UM MODELO DE BIBLIOTECA VIRTUAL

Com as tecnologias acima descritas, é possível definir um modelo para a implementação de um ambiente *peer-to-peer* para uma biblioteca virtual que proporcione documentos autênticos, com segurança e confiabilidade na sua integridade.

A biblioteca virtual pode ser considerada um serviço na web, pois provê aos usuários o acesso a material científico através do acesso remoto. Por ser considerado um serviço web, deve seguir alguns conceitos para ser bem sucedido.

3.1 Segurança

Uma parte fundamental de quaisquer serviços na web é a segurança, esta pode assumir várias formas. Em especial para sistemas distribuídos, como é o caso do *peer-to-peer*, necessitamos de quatro formas de segurança básicas:

- Confidencialidade: proteger a informação trocada de acessos não autorizados;
- Integridade: apenas os autorizados podem modificar a informação;
- Autenticidade: identificar que a origem e o destino da mensagem não são falsos;
- Disponibilidade: os serviços devem estar sempre disponíveis a uma margem de usuários.

3.1.1 Segurança no JXTA

A segurança no JXTA se dá com base no modelo de segurança por papel, no qual cada nodo individual opera sobre a autoridade concedida por outro nodo confiável para executar uma determinada tarefa [JXTA 2003].

O JXTA suporta cinco formas de segurança, sendo que três delas mencionadas acima (Confidencialidade, Integridade, Autenticidade), e duas outras:

- Autorização: garante que o remetente é autorizado ao envio da mensagem;
- Refutabilidade: garante que a mensagem foi transmitida por um remetente devidamente identificado e não é uma re-transmissão de uma mensagem anterior.

As seguranças garantidas pelo JXTA são possíveis graças às mensagens XML, que permitem a inclusão de metadados. Cada metadado garante um aspecto da segurança necessária, resumos de mensagem (hash) garantem a integridade da mensagem, criptografia (com chave pública) e assinaturas digitais (com certificados) garantem confidencialidade e refutabilidade, e credenciais podem ser usadas para autenticação e autorização de mensagens [JXTA 2003].

O JXTA suporta SSL²⁴ (*Secure Sockets Layer*) e IPSec²⁵ (*Internet Protocol Security*), entretanto estes protocolos providenciam somente confidencialidade e integridade da mensagem.

²⁴ Um protocolo desenvolvido pela Netscape para a transmissão de conteúdo particular pela internet. Funciona através do uso de uma chave privada para a criptografia dos dados transmitidos pela conexão SSL [WEBOPIDIA 2004].

²⁵ Um conjunto de protocolos desenvolvido pela IETF para suportar a troca segura de pacotes na camada IP. Usado na implementação de VPNs (*Virtual Private Networks*) [WEBOPIDIA 2004].

3.1.2 Segurança para a biblioteca virtual

O modelo de uma fonte de conhecimento segura, deve abranger mais do que as cinco formas de seguranças vistas. Além de prover a segurança dos dados trafegados e de quem está navegando na rede, e necessário prover também a segurança do conteúdo da rede.

Por causa da natureza do software se faz necessária que algumas medidas de segurança sejam aplicadas em cima dos nodos que compartilham os dados da rede, estes métodos de segurança variam com o estilo da rede.

Quando a rede é denominada privada e o conteúdo da rede é restrito a uma ou mais instituições que desejam que esse conteúdo seja apenas aquele publicado pelas suas autoridades delegadas, é utilizado um modelo de nodo autoridade que restringe o conteúdo da rede, visto na seção 3.1.2.1.

Quando a rede é denominada pública, mas ainda é necessário que o conteúdo publicado pelas autoridades da rede seja diferenciado, é utilizado um modelo de nodos credenciadores, que identificam conteúdo publicado pelas autoridades delegadas, visto na seção 3.1.2.2.

3.1.2.1 Nodos autoridades

Como o conteúdo da rede são documentos autênticos, é necessário que a publicação de arquivos na rede seja permitida para apenas documentos autênticos legítimos. Para isso é utilizada a idéia de autoridades dentro da rede, aonde somente essas autoridades podem publicar novos materiais. A figura 3.1.2.1a ilustra uma rede *peer-to-peer* com um nodo autoridade:

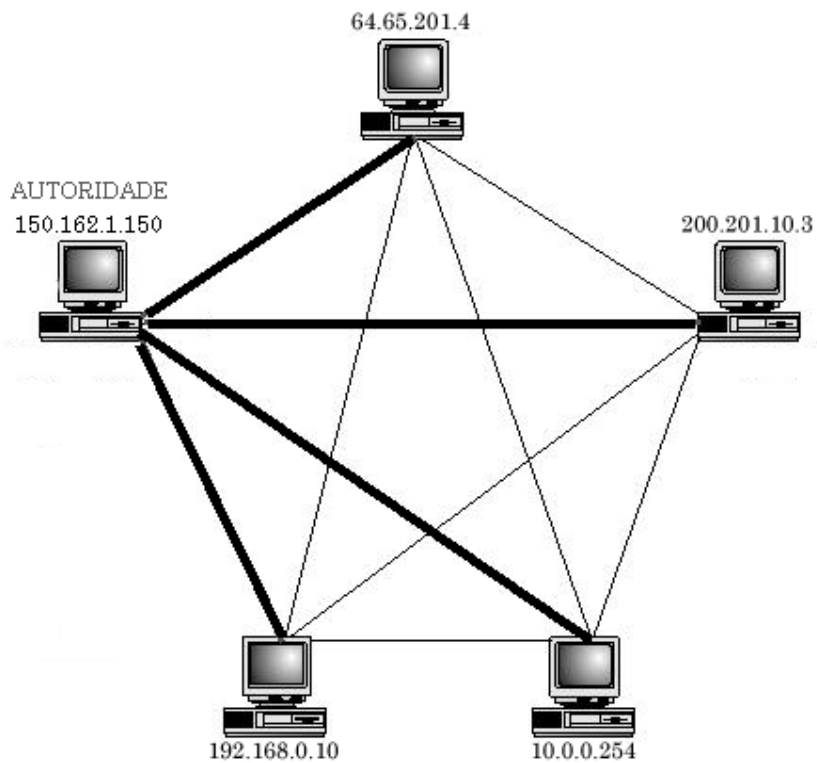


Figura 3.1.2.1a – Rede *peer-to-peer* com nó autoridade

Nodos autoridade, possuem a função de publicar conteúdo novo na rede, e autenticar a publicação de conteúdo por nodos normais. Os nodos normais têm permissão para publicar somente conteúdo já publicado pelos nodos autoridade e que estão presentes na rede, assim garantindo que os dados da rede são legítimos e confiáveis.

Os nodos autoridades, na concepção da biblioteca virtual, devem ser constituídos somente das instituições responsáveis pelo conteúdo da rede.

A autenticação de conteúdo é dada através do armazenamento de resumos (hash) dos arquivos, e de uma criptografia de chave pública para trafegar as mensagens de pedido de publicação de forma segura. A figura 3.1.2.1b ilustra o processo de publicação de conteúdo de um nodo normal.

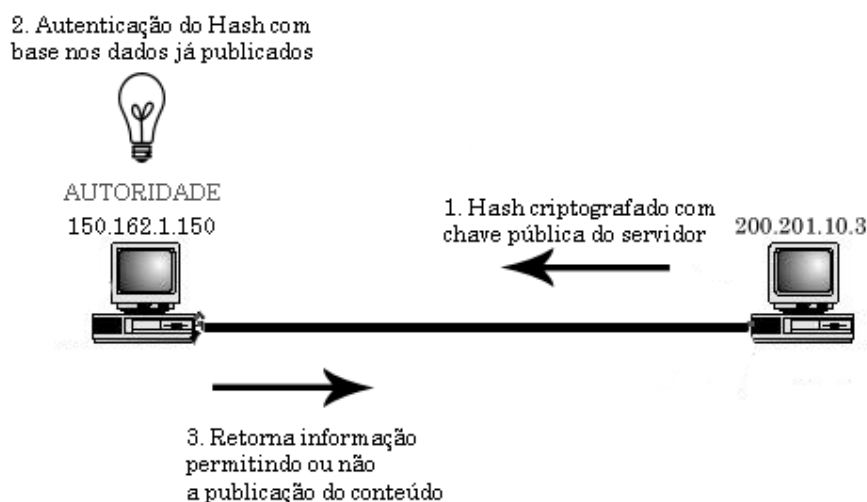


Figura 3.1.2.1b – Esquema de publicação

3.1.2.2 Nodos credenciadores

Quando o objetivo da rede é ser totalmente pública, ou seja permitir que os usuários contribuam com a rede com quaisquer tipos de documentos, a fim de enriquecer a rede, é utilizada uma abordagem similar a vista na seção 3.1.2.1.

Nodos credenciadores são baseados nos nodos autoridades, entretanto não exercem a função de permitir ou não a publicação de conteúdo na rede. Os nodos credenciadores têm como função básica diferenciar o conteúdo credenciado pelas autoridades, e o conteúdo publicado pelos usuários da rede.

Outras funções de um nodo credenciador são a publicação de conteúdo na rede e o credenciamento de conteúdo da rede. O credenciamento de conteúdo pode ser feito a qualquer conteúdo publicado na rede, mesmo que esse conteúdo não tenha sido publicado por um nodo credenciador, desde que o conteúdo já não esteja credenciado.

Essa diferenciação de conteúdo garante que os documentos credenciados pelos nodos credenciadores sejam assegurados, e mesmo assim permite que a rede

suporte a adição de novos documentos pelos usuários da rede, a fim de enriquecer o conteúdo da rede com a contribuição dos usuários.

3.2 QoS

Qualidade de Serviço (*Quality of Service*), ou QoS, se tornou um fator crítico para o sucesso de redes empresariais. Com o aumento da banda larga, a afirmação que a largura de banda é tão abundante que o QoS não se torna necessário é falsa [TECHGUIDE 2001].

Qualidade de Serviço é um termo de redes que especifica um nível aceitável de transmissão de dados [WEBOPIDIA 2004], e pode trabalhar em vários níveis.

3.2.1 QoS em um ambiente *peer-to-peer*

O QoS pode se manifestar de várias formas, entretanto algumas funcionalidades são úteis para um ambiente *peer-to-peer*, elas são:

- Balanceamento de carga: Distribui a carga de rede através dos nodos que possuem a informação necessária;
- Repositórios Dinâmicos de Informação: Dada à natureza de redes *peer-to-peer*, quanto mais procurado um dado, mais ele se espalha pela rede, e mais ele se torna disponível;
- Redundância e Tolerância à falhas: Replicando a informação em diversos nodos cria um alto nível de redundância, logo uma alta disponibilidade, e assim diminui a chance de falha, uma vez que a saída de um nodo não causa muito impacto na rede.

Em redes *peer-to-peer* de trocas de arquivos não existe somente o QoS da rede que trafega a informação, mas também da informação em si. Normalmente utilizado um sistema de qualificação dos arquivos, para determinar a qualidade do

arquivo dentro da rede. Esse tipo de QoS, entretanto, não se aplica as redes que usam dos nodos de autoridade, que publicam apenas documentos autênticos na rede. Entretanto esse tipo de QoS pode ser aplicado para as redes que usam dos nodos credenciadores, a fim de possibilitar aos usuários a qualificação de conteúdo publicado por outros usuários.

3.2.2 QoS aplicado à biblioteca virtual

Devido à estrutura da rede *peer-to-peer* da biblioteca virtual, é necessário especificar como cada métrica de QoS será aplicada a esta rede híbrida.

3.2.2.1 Balanceamento de carga (Rede)

A rede híbrida da biblioteca com nodos autoridades possui pontos centralizados de conteúdo, nesses nodos a grande maioria do conteúdo da rede estará focada. Uma vez que os outros nodos da rede adquirem certa quantidade de conteúdo válido, e o compartilham, o balanceamento de carga se torna mais eficiente, tirando um pouco da responsabilidade dos nodos autoridades, e distribuindo a banda entre os outros nodos da rede [Figura 3.2.2.1a].

Quando é utilizado o modelo com nodos credenciadores, o conteúdo da rede não é totalmente centralizado, e o balanceamento de carga se dá à medida que quaisquer conteúdos são distribuídos entre os usuários. Quanto mais usuários possuírem um documento, maior é o balanceamento de carga para esse documento em específico.

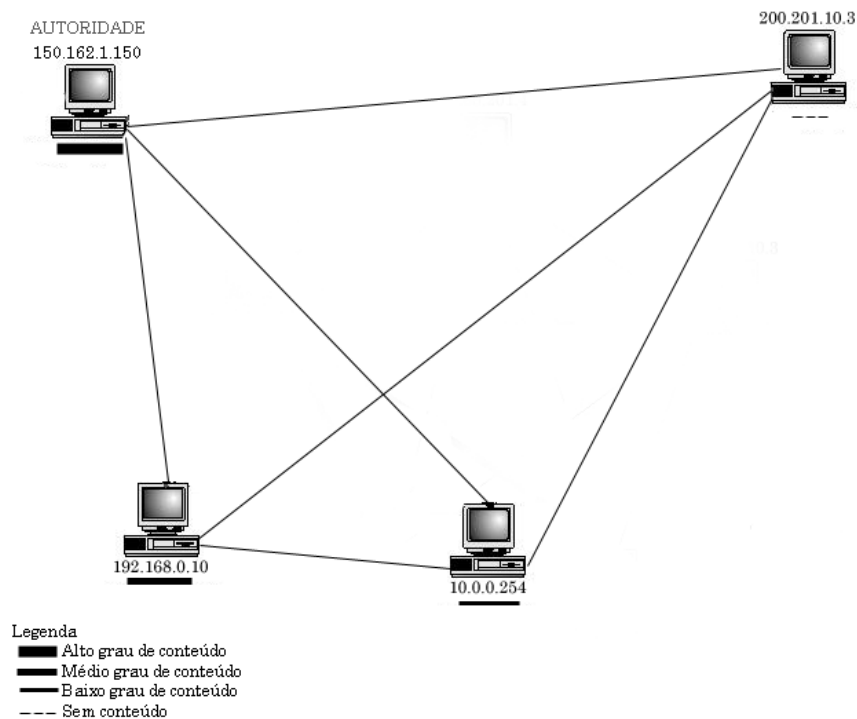


Figura 3.2.2.1a – Nível de conteúdo

3.2.2.2 Repositórios dinâmicos de informação

O repositório de dados da rede funciona da mesma forma, e se espalha com o tempo, com a única diferença que quando utilizado o modelo com nodos autoridades os novos arquivos só podem ser publicados em um local, assim o foco de origem de conteúdo novo na rede se dá em poucos pontos, mas tende-se a espalhar pela rede da mesma forma.

Desde que o conteúdo que os nodos adquiram não sofra alterações que possam comprometer seus dados, modificarem o nome do arquivo não é uma dessas modificações, esse nodo se torna um ponto de distribuição desse conteúdo obtido, e mais um fornecedor de conteúdo na rede.

A mudança desse documento faz com que ele se torne um novo documento, em redes com a visão de nodos autoridades, esse documento modificado não mais

poderá ser circulado na rede, garantindo a unicidade de documentos. Em redes com a visão de nodos credenciadores, o documento pode ser publicado na rede, entretanto não é visto como um documento credenciado, e assim pode ser detectada a alteração do conteúdo do documento.

3.2.2.3 Redundância e tolerância à falhas

Devido à natureza híbrida e semi-centralizada da rede planejada, há uma diminuição na tolerância à falhas, já que a queda de um nodo autoridade diminui em muito o conteúdo da rede e a queda de um nodo credenciador diminui a segurança quanto à autenticidade do conteúdo da rede. Entretanto, à medida que os nodos se tornam mais participativos dentro da rede, a robustez aumenta.

Podemos concluir que quanto mais um conteúdo específico for requisitado, mais ele estará disponível.

3.2.2.4 Balanceamento de carga (Cliente)

Uma abordagem vista em softwares *peer-to-peer* atuais, é o uso de um controle de velocidade de transferência, para evitar uma sobrecarga da largura de banda, e o gasto exaustivo de recursos.

Este princípio parte do pressuposto que a largura de banda deve ser compartilhada com outros serviços concorrentes no cliente, e deve permitir que sejam configurados níveis aceitáveis de desempenho, para cada usuário, a fim de não consumir todos os recursos da máquina.

4. ANÁLISE DO MODELO DE BIBLIOTECA VIRTUAL PARA UMA BIBLIOTECA VIRTUAL PÚBLICA

Usando da análise UML, é possível modelar o funcionamento de uma ferramenta, nesse caso, não só a ferramenta em si, mas a rede *peer-to-peer* para uma biblioteca virtual pública.

A biblioteca é um sistema distribuído, que contém dois tipos de componentes, os softwares para usuários comuns da biblioteca, e os nodos credenciadores.

O usuário comum dessa biblioteca pública tem como objetivo base pesquisar informação na rede, e fazer o download dessa informação. Por se tratar de uma rede *peer-to-peer*, o usuário pode contribuir para com a rede, através da publicação de documentos novos ou já contidos na rede, de forma a criar uma redundância e balanceamento de carga da informação compartilhada na rede.

O nodo credenciador, por sua vez, tem como objetivo base, fornecer conteúdo para a rede, e assegurar que o conteúdo da rede marcado como credenciado é seguro e confiável. O nodo credenciador deve assegurar que os documentos credenciados não tenham seu conteúdo alterado.

4.1 Visão dos atores

Através de um diagrama de caso de uso, é possível mostrar as visões que cada tipo de ator do sistema tem sobre os casos de uso, os quais são representados pelas elipses [Figura 4.1a].

Cada caso de uso contido dentro desse, possui uma documentação que possibilita o entendimento do funcionamento do sistema, com base nas suas funções principais como veremos mais adiante nas próximas seções.

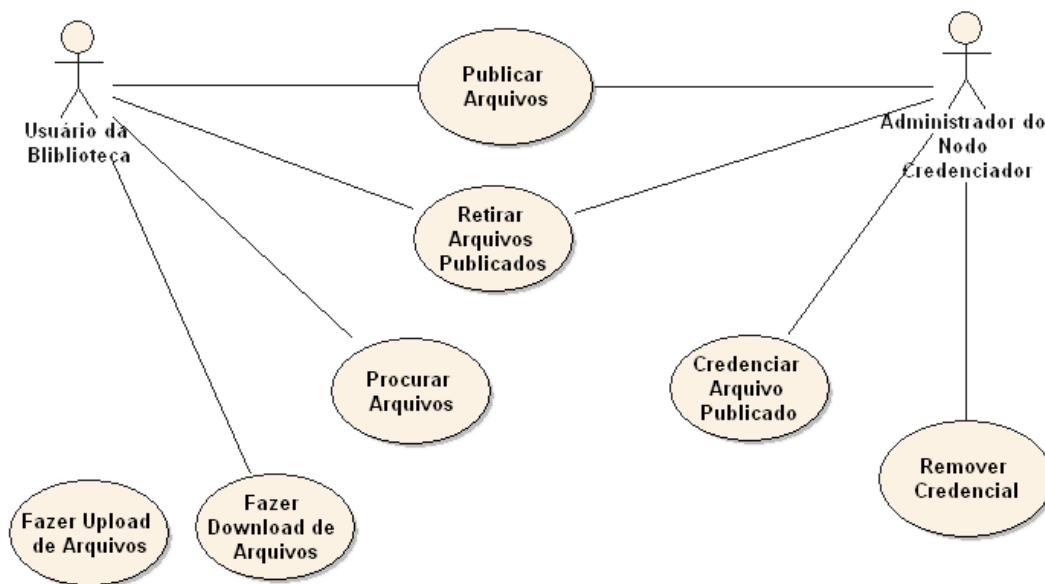


Figura 4.1a – Diagrama de Casos de Uso

4.2 Funcionamento da rede

A rede *peer-to-peer* da biblioteca virtual é considerada um rede híbrida, pois para ser feita a localização da informação deve-se autenticar os elementos da procura nos nodos credenciadores, que agem como índices centrais para informação credenciada na rede.

4.2.1 Indexação centralizada

O problema com a centralização de índices, é que a rede se torna dependente dos nodos credenciadores, assim tornando a rede suscetível a ataques, uma vez que o seu funcionamento correto depende de elementos específicos como os nodos credenciadores, caso esses não estejam disponíveis, não é possível diferenciar arquivos credenciados publicados pelas autoridades e arquivos publicados por usuários.

É possível amenizar esse problema aumentando o número de nodos credenciadores, quanto maior o número deste tipo de nodos, menos chance da autenticação do conteúdo credenciado na rede ser interrompido.

Outra medida para garantir que a queda de um dos nodos credenciado não cause perda da validação de parte do conteúdo na rede, é o compartilhamento do índice de arquivos credenciados entre nodos credenciadores, assim é possível que um nodo credenciador valide arquivos credenciados por outro nodo credenciador.

4.2.2 Segurança de conteúdo

O problema de segurança ocasionado pela indexação centralizada gera outros benefícios, que são vitais para o funcionamento da rede. O maior deles é a verificação dos arquivos que circulam por ela. Essa verificação (validação) é necessária para garantir que o conteúdo credenciado na rede seja apenas aquele credenciado pelos nodos credenciadores, e que esse mesmo conteúdo não seja modificado.

Isso é possível através da criação de identificadores únicos para cada arquivo publicado, e pode ser obtido através do hash do arquivo. O hash do arquivo para a geração de um identificador único já é utilizado em programas *peer-to-peer* existentes, a idéia é aproveitar esse identificador para garantir que o conteúdo da rede não seja alterado por pessoas não autorizadas a fazer o mesmo.

Uma vez que um arquivo publicado na rede seja alterado localmente, o hash desse arquivo será diferente do arquivo original, assim ele não poderá ser distribuído na rede como um arquivo credenciado, e, portanto, a originalidade do documento será mantida dentro da rede.

No caso da biblioteca proposta, o conteúdo seriam produções científicas públicas, e os órgãos responsáveis pelos nodos autoridade, seriam as faculdades federais e estaduais participantes da rede.

4.3 Funcionalidades básicas

Qualquer programa *peer-to-peer* de troca de arquivo, possui três funcionalidades básicas, sendo elas, pesquisa por conteúdo, download de conteúdo e compartilhamento de conteúdo. A biblioteca virtual pública não difere nesse aspecto dos programas de troca de arquivo, entretanto, essas funcionalidades devem ser alteradas a fim de incluir os passos para autenticar os arquivos com os nodos credenciadores.

4.3.1 Pesquisa

A pesquisa para a biblioteca virtual não difere das pesquisas por arquivos de redes *peer-to-peer* comuns. Para ser feita uma pesquisa, é necessário que seja enviado um conjunto de critérios de pesquisa para os vários nodos da rede, a fim de se receber um conjunto de arquivos como resposta.

Esses critérios são os metadados associados com os arquivos publicados. Para cada tipo de arquivo publicado na rede, deve haver um conjunto de metadados que servirão como campos de pesquisa para o arquivo.

Um passo a mais é a requisição das credenciais de cada arquivo mostrado na resposta, a fim de diferenciar os arquivos credenciados dos não credenciados.

4.3.2 Download

O download de documentos na rede é feito através da requisição direta ao nodo que contém o documento, uma vez que esse arquivo seja selecionado de uma lista de documentos populada com uma pesquisa bem sucedida.

4.3.3 Compartilhamento

No compartilhamento de documentos, o usuário apenas seleciona o documento ao qual deseja compartilhar com a rede, e este documento é incluso na rede da biblioteca.

4.4 Funcionalidades específicas

O nodo credenciador possui algumas funções específicas às quais só podem ser executadas por ele. Uma dessas funções é a de servir como principal fonte de conteúdo para a rede. Outra função é de agir como uma autoridade de certificações, validando documentos que circulam pela rede.

4.5 Descrição do modelo proposto

Nesta seção busca-se relatar de modo genérico os processos dinâmicos que representam o modelo da biblioteca virtual aqui proposto, descrevendo para tal os casos de uso expandidos: da procura de arquivos, do modo de fazer *download* de arquivos, da publicação de arquivos e do modo de retirar arquivos publicados, demonstrando a dinâmica desses processos através de diagrama de seqüências e definição de contratos.

4.5.1 Caso de uso expandido: Procura de Arquivos

Com a finalidade de procurar arquivos através de dados fornecidos, definimos como atores deste caso de uso os usuários do sistema da biblioteca virtual (Figura 4.5.1a), onde o mesmo preenche dados referentes às especificações da produção científica desejada e inicia o processo de busca na rede arquivos que se encaixem nos dados procurados.

Atores : Usuário da Biblioteca

Tipo: Primário e Essencial.

Seção Principal – Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
1. O usuário entra os dados necessários para efetuar a pesquisa.	
2. O usuário manda efetuar a busca	3. O sistema coloca as informações digitadas no formato RDF/XML e envia um pedido a rede.
	4. O sistema retorna uma lista de arquivos que se encaixam nos critérios de busca definidos.
	5. O sistema envia a lista de identificadores dos arquivos para o nodo certificador mais perto.
	6. O nodo certificador devolve ao sistema a lista dos arquivos credenciados, com as informações das credenciais.

Seqüências Alternativas:

Linha 1: O usuário pode clicar no botão limpar e recomeçar o caso de uso;

Linha 4: O sistema não encontra arquivos que se encaixam nos critérios de busca definidos.

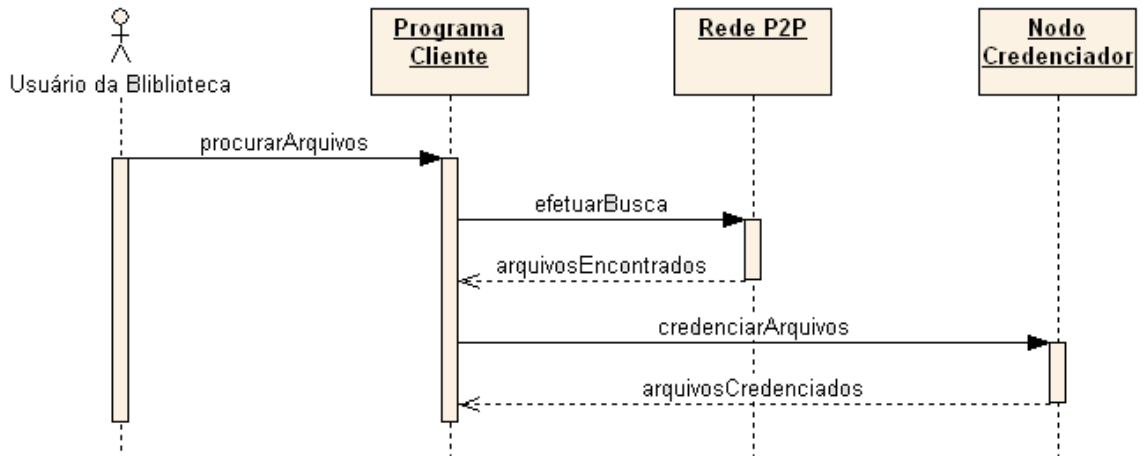


Figura 4.5.1a – Diagrama de seqüência do caso de uso da procura de arquivos

Contratos

- **Contrato:** procurarArquivos
- **Responsabilidades:** Procurar um arquivo na rede.
- **Tipo:** Sistema.
- **Referências Cruzadas:** Caso de uso Procurar Arquivos.
- **Notas:** A procura autentica as credenciais da lista dos arquivos.
- **Exceções:** Nenhum arquivo localizado com os critérios procurados.
- **Saída:** Lista de arquivos.
- **Pré-condições:** Arquivo que se encaixe nos critérios de busca e deve estar publicado na rede.
- **Pós-condições:** Se não existia uma lista de arquivos, uma lista foi criada; Se uma lista de arquivos existia, ela foi sobrescrita.

4.5.2 Caso de uso expandido: Fazer Download de Arquivos

Neste caso o usuário seleciona um arquivo de uma lista de procura, e pede para o sistema efetuar o download do arquivo, conforme descreve a Figura 4.5.2a.

Atores: Usuário da Biblioteca

Tipo: Primário e Essencial.

Seção Principal – Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
1. O usuário seleciona um arquivo da lista de arquivos pesquisados.	
	2. O sistema contacta as localizações do arquivo e abre canais de transferência.
	3. O arquivo chega ao sistema e é armazenado.

Seqüências Alternativas:

Linha 1: O usuário pode clicar no botão pesquisar e recomeçar o caso de uso;

Linha 2: O sistema não consegue contatar os locais aonde o arquivo se encontra, repetir 3 vezes, depois finalizar use case;

Linha 3: Problemas na transferência do arquivo, voltar à linha 2.

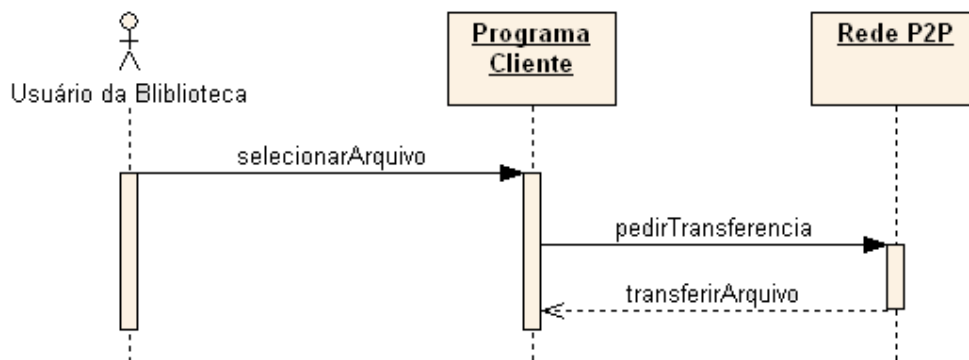


Figura 4.5.1a – Diagrama de seqüência do caso de: Fazer download de arquivos

Contratos

- **Contrato:** `selecionarArquivo`
- **Responsabilidades:** Seleciona um arquivo para efetuar download.
- **Tipo:** Sistema.
- **Referências Cruzadas:** Caso de uso Procurar Arquivos; Caso de uso Fazer Download de Arquivos.
- **Notas:** Depende da disponibilidade do arquivo, caso a mesmo não possa ser encontrado, não é possível efetuar o download do arquivo.

- **Exceções:** O arquivo não é localizado.
- **Saída:** Arquivo gerado em pasta local no sistema
- **Pré-condições:** Uma lista de arquivos populada;
- **Pós-condições:** Se o arquivo não existia, um arquivo foi criado.

4.5.3 Caso de uso expandido: Publicar Arquivos

Com a finalidade de publicar um arquivo na rede da Biblioteca, o usuário seleciona um arquivo de uma pasta de arquivos compartilhados. É gerado o hash do arquivo. O arquivo é publicado na rede (Figura 4.5.3a).

Atores: Usuário da Biblioteca, Administrador do Nodo Credenciador.

Tipo: Primário e Essencial.

Seção Principal – Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
1. O usuário seleciona um arquivo de uma pasta de arquivos compartilhados, e solicita a publicação do mesmo.	
	2. O sistema gera o hash do arquivo, para identificá-lo na rede.
	3. O sistema valida o arquivo como compartilhado e publicado na rede.

Seqüências Alternativas:

Linha 1: O usuário pode clicar no botão cancelar, fim do caso de uso;

Linha 3: O arquivo já está publicado nesse nodo, o sistema aborta a publicação do arquivo e gera um log de erro.

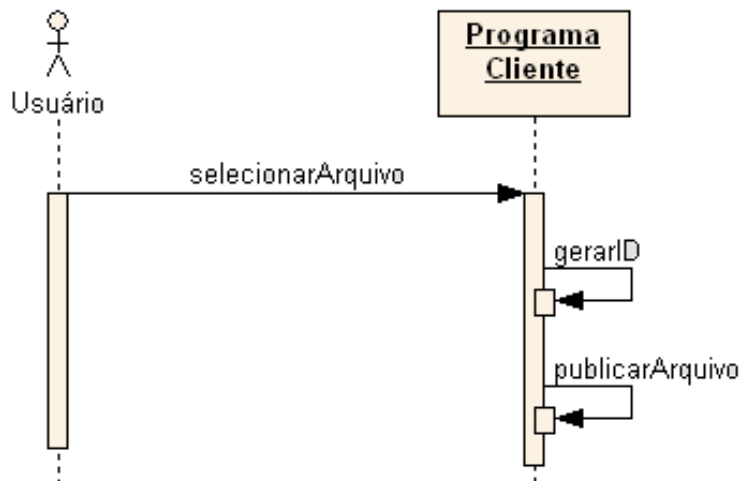


Figura 4.53a – Diagrama de seqüência do caso de uso: Publicar Arquivos

Contratos

- **Contrato:** `selecionarArquivo`
- **Responsabilidades:** Seleciona um arquivo para ser publicado na rede.
- **Tipo:** Sistema.
- **Referências Cruzadas:** Caso de uso Publicar Arquivos.
- **Notas:**
- **Exceções:** Arquivo já publicado.
- **Saída:**
- **Pré-condições:** Um arquivo disponível para publicação, dos tipos aceitos pela rede.
- **Pós-condições:** Se o Arquivo não foi encontrado, o arquivo foi publicado.

4.5.4 Caso de uso expandido: Retirar Arquivo Publicado

No caso de como retirar um arquivo publicado demonstra o usuário selecionando um arquivo publicado na rede e o sistema retirando o compartilhamento do arquivo da rede, conforme Figura 4.5.4a.

- **Atores:** Usuário da Biblioteca, Administrador do Nodo Credenciador.

- **Tipo:** Secundário.

Seção Principal – Seqüência Típica de Eventos – Usuário da Biblioteca

Ação do Ator	Resposta do Sistema
1. O usuário seleciona um arquivo compartilhado e seleciona a opção de remover o compartilhamento.	
	2. O sistema retira o arquivo da lista de arquivos compartilhados.

Seqüências Alternativas:

Linha 1: O usuário pode clicar no botão cancelar, fim do caso de uso.

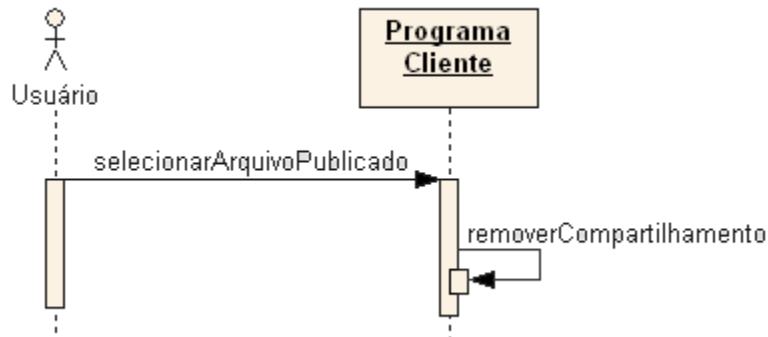


Figura 4.5.4a – Diagrama de seqüência do caso de uso: Retirar Arquivo Publicado

Contratos

- **Contrato:** `selecionarArquivoPublicado`
- **Responsabilidades:** Seleciona um arquivo publicado localmente.
- **Tipo:** Sistema.
- **Referências Cruzadas:** Caso de uso Retirar Arquivo Publicado.
- **Notas:**
- **Exceções:** Nenhum arquivo publicado.
- **Saída:**
- **Pré-condições:** Um arquivo já publicado no programa local.
- **Pós-condições:** Se o Arquivo foi encontrado, foi removido da lista de arquivos publicados.

4.5.5 Caso de uso expandido: Credenciar Arquivo Publicado

Nesse caso de uso, o usuário do nodo credenciador seleciona um arquivo publicado na rede, ou local (Figura 4.5.5a) ou remoto (Figura 4.5.5b), a fim de criar um credenciamento para o arquivo, e publicá-lo nos nodos credenciadores.

- **Atores:** Administrador do Nodo Credenciador.
- **Tipo:** Primário e Essencial.

Seção Principal – Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
1. O usuário seleciona um arquivo compartilhado e seleciona a opção de credenciar o arquivo.	
	2. O sistema verifica se o arquivo não possui credenciais em outros nodos autoriade.
	3. O sistema gera as credenciais para o arquivo.
	4. O sistema atualiza sua lista de credenciais e envia a lista atualizada para os nodos credenciadores disponíveis.

Seqüências Alternativas:

Linha 1: O usuário pode clicar no botão cancelar, fim do caso de uso;

Linha 2: O arquivo já está credenciado por outro nodo credenciador, fim do caso de uso.

Seção Secundária – Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
1. O usuário entra os dados necessários para efetuar a pesquisa.	
2. O usuário manda efetuar a busca	3. O sistema coloca as informações digitadas no formato RDF/XML e envia um pedido a rede.
	4. O sistema retorna uma lista de arquivos que se encaixam nos critérios de busca definidos.
	5. O sistema envia a lista de identificadores dos arquivos para o nodo certificador mais perto.
	6. O nodo certificador devolve ao sistema a lista dos arquivos credenciados, com as informações das credenciais.
7. O usuário seleciona um arquivo ainda não	

credenciado da lista.	
	8. O sistema gera as credenciais para o arquivo.
	9. O sistema atualiza sua lista de credenciais e envia a lista atualizada para os nodos credenciadores disponíveis.

Seqüências Alternativas:

Linha 1 e 7: O usuário pode clicar no botão cancelar, fim do caso de uso;

Linha 4: O sistema não encontra arquivos que se encaixam nos critérios de busca definidos, fim do caso de uso.

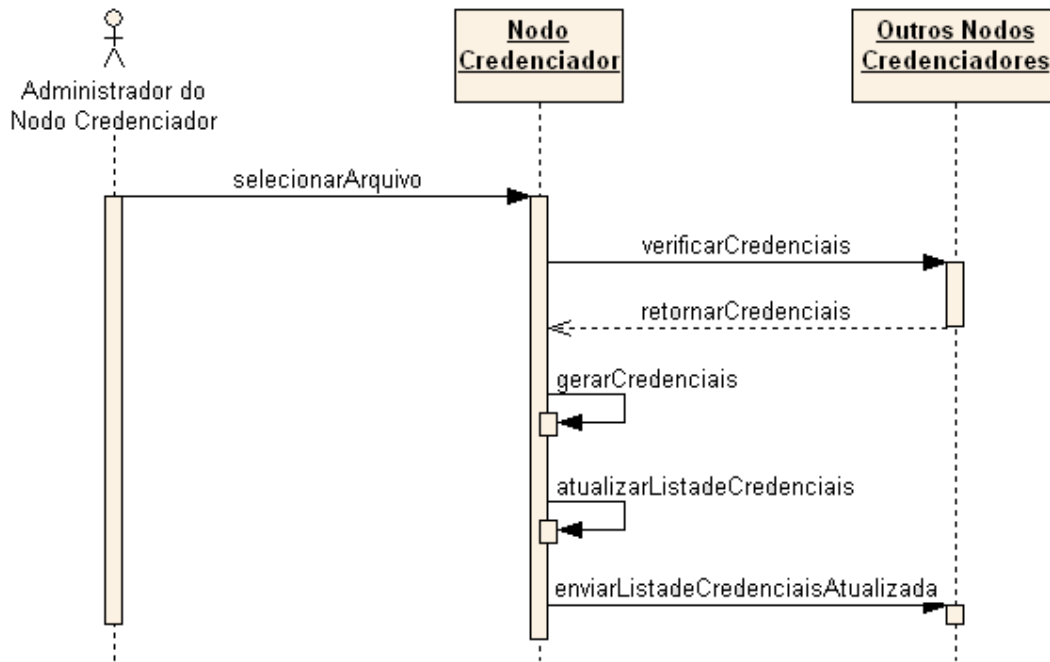


Figura 4.5.5a – Diagrama de seqüência principal do caso de uso: Credenciar Arquivo Publicado

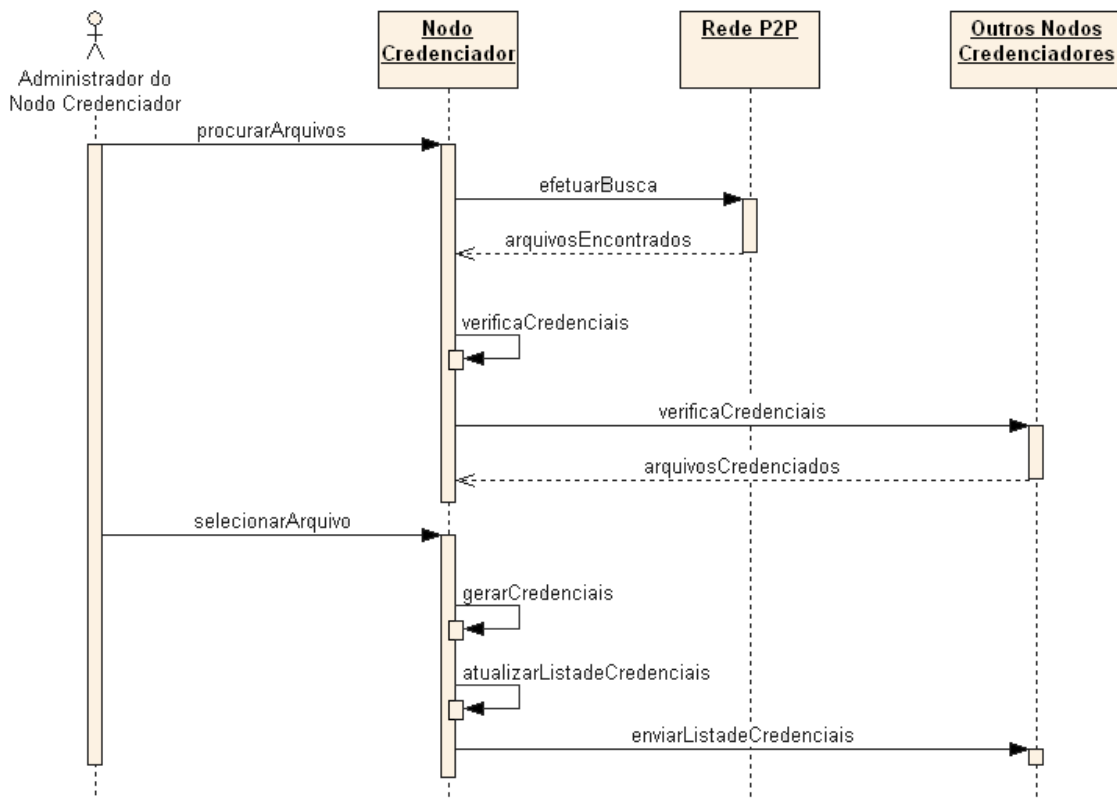


Figura 4.5.5b – Diagrama de seqüência secundária do caso de uso: Credenciar Arquivo Publicado

Contratos

- **Contrato: selecionarArquivo**
- **Responsabilidades:** Selecionar um arquivo publicado sem credenciais.
- **Tipo:** Sistema.
- **Referências Cruzadas:** Caso de uso Credenciar Arquivo Publicado;
Contrato procurarArquivos.
- **Notas:** Se o arquivo não for local, deve-se usar a seqüência secundária de passos.
- **Exceções:** Nenhum arquivo publicado.
- **Saída:** Credencial.
- **Pré-condições:** Um arquivo já publicado e sem credenciais.
- **Pós-condições:** Se o arquivo foi encontrado, foi criada uma credencial para o arquivo.

- **Contrato:** procurarArquivo
- **Responsabilidades:** Procurar um arquivo sem credenciais na rede.
- **Tipo:** Sistema.
- **Referências Cruzadas:** Caso de uso Credenciar Arquivo Publicado;
Contrato selecionarArquivo.
- **Notas:** A procura verifica se o arquivo possui credenciais, e somente mostra os arquivos sem credenciais.
- **Exceções:** Nenhum arquivo sem credenciais localizado com os critérios procurados.
- **Saída:** Lista de arquivos.
- **Pré-condições:** Arquivo que se encaixe nos critérios de busca, deve estar publicado na rede e estar sem credenciais.
- **Pós-condições:** Se não existia uma lista de arquivos, uma lista foi criada; Se uma lista de arquivos existia, ela foi sobrescrita.

4.5.6 Caso de uso expandido: Remover Credencial

O administrador do nodo credenciador seleciona uma credencial de um arquivo da rede que tenha sido criado por ele, remove essa credencial de sua lista de credenciais, e atualiza a sua lista de credenciais com os outros nodos credenciadores na rede (Figura 4.5.6a).

- **Atores:** Administrador do Nodo Credenciador.
- **Tipo:** Secundário.

Seção Principal – Seqüência Típica de Eventos

Ação do Ator	Resposta do Sistema
1. O usuário seleciona uma credencial na lista de credencias do nodo credenciador, e seleciona a opção de remover a credencial.	
	2. O sistema apaga a credencial e atualiza sua lista de credenciais.
	3. O sistema envia sua lista de credenciais atualizada para os nodos credenciadores disponíveis.

Seqüências Alternativas:

Linha 1: O usuário pode clicar no botão cancelar, fim do caso de uso;

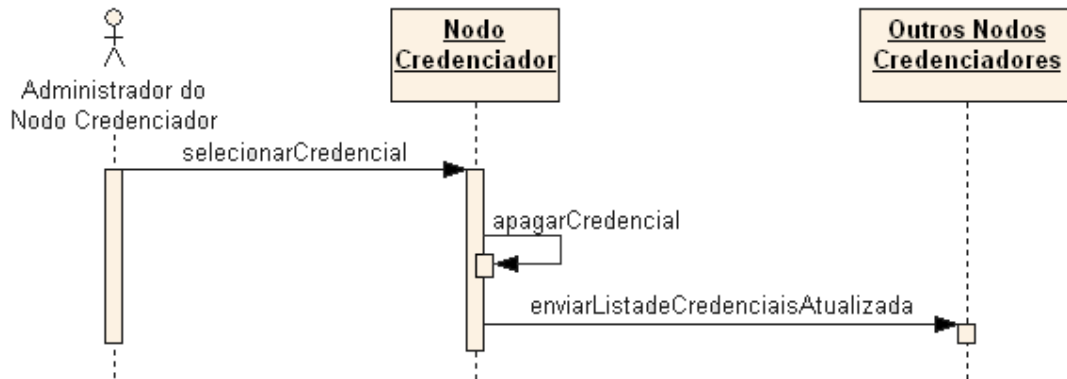


Figura 4.5.6a – Diagrama de seqüência do caso de uso: Remover Credencial

Contratos

- **Contrato:** selecionarCredencial
- **Responsabilidades:** Selecionar uma credencial a ser removida da lista de credenciais.
- **Tipo:** Sistema.
- **Referências Cruzadas:** Caso de uso Remover Credencial.
- **Notas:**
- **Exceções:** Nenhuma credencial disponível.
- **Saída:**
- **Pré-condições:** Credencial deve existir.
- **Pós-condições:** Credencial removida da lista de credenciais, lista de credenciais atualizada nos nodos credenciadores.

4.6 Modelo Conceitual

Utilizando como referência o framework JXTA, é proposto um modelo conceitual de implementação das principais classes para a biblioteca virtual (Figura

4.6a), esse modelo mostra a classe pai **Nodo** que guarda os atributos e métodos comuns a todos os tipos de nodos, também é mostrado o framework **JXTA**, sendo que muitos tipos utilizados no modelo são provindos desse framework.

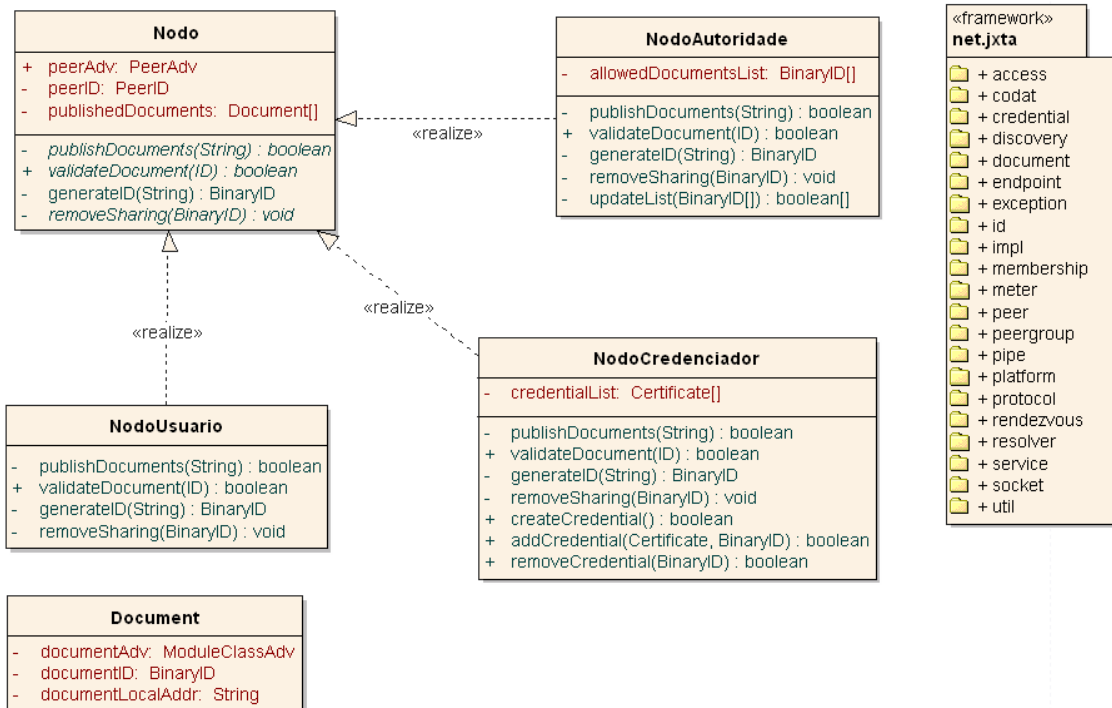


Figura 4.6a – Modelo Conceitual da Biblioteca Virtual

5. CONCLUSÕES E TRABALHOS FUTUROS

5.1 Conclusão

Muito embora este trabalho de pesquisa trate apenas de um modelo analítico, pode-se perceber que a catalogação do conteúdo de uma rede *peer-to-peer* com o uso da indexação do conteúdo da rede, e da autenticação dos arquivos compartilhados por nodos autoridades ou o credenciamento por nodos credenciadores, é, teoricamente, possível garantir que a rede possua dados publicados e autenticados pelas entidades responsáveis pelo conteúdo, total ou parcialmente, dependendo do modelo adotado.

Verificou-se também que através do nodo autoridade podemos garantir que o conteúdo da rede seja apenas aquele autorizado a ser compartilhado, assim tornando possível o uso de redes *peer-to-peer* para a distribuição de arquivos controlados, por um conjunto de autoridades certificadoras.

Por fim, conclui-se que uma aplicação de biblioteca virtual, conforme modelo proposto no capítulo 3 e 4 deste trabalho, propõe-se a gerar uma rica fonte de conhecimento para acadêmicos e pesquisadores, podendo ser de fonte confiável, mesmo que obtido digitalmente, e não restringe os pesquisadores ao conteúdo publicado pelas autoridades, mas permite o compartilhamento da informação entre comunidades de pesquisadores.

5.2 Trabalhos futuros

A implementação desse modelo de biblioteca virtual, proposto na seção 4, seria um plano para o futuro, para que possa ser validado o modelo sugerido, desde

que autoridades pudessem fornecer conteúdo para a rede e estivessem dispostos a participar.

Com a implementação do modelo de biblioteca virtual, pode ser feito uma análise comparativa para determinar o impacto dos nodos autoridade em redes *peer-to-peer* sem indexação centralizada de conteúdo com validação de conteúdo, a fim de verificar se o impacto dos nodos autoridade ou credenciadores justifica sua confiabilidade sobre o conteúdo da rede.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [CERVO 1996] CERVO, Amando Luiz; BERVIAM, Pedro Alcino. **Metodologia Científica**. 4º Edição. São Paulo: Makron Books, 1996. p.20-22,44-45,48-48.
- [FALSHIN 1993] FALSHIN, Odília. **Fundamentos de Metodologia**. São Paulo: Atlas, 1993. p. 36-38,101-104.
- [GIL 1996] GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 3ª Ed. São Paulo: Editora Atlas, 1996.
- [JXTA 2003] **Project JXTA v 2.0**: Java Programmers Guide. Sun Microsystems, Maio de 2003. Disponível em http://www.jxta.org/docs/JxtaProgGuide_v2.pdf. Acessado em: 5 de Julho de 2004.
- [ORAM 2001] ORAM, Andy. **PEER-TO-PEER**: O poder transformador das redes ponto a ponto. São Paulo: Editora Berkeley, 2001.
- [SALOMON 1978] SALOMON, Délcio Vieira. **Como fazer uma Monografia de trabalho científico**. Belo Horizonte: Interlivros, 1978. p.137-138,140-147.
- [SILVA 2000] SILVA, R. P. **Suporte ao desenvolvimento e uso de frameworks e componentes**. Tese de doutorado. Porto Alegre: UFRGS/II/PPGC, mar. 2000. 262p. Disponível em: <http://www.inf.ufsc.br/~ricardo/download/tese.pdf>. Acesso em: 12 de junho de 2004.
- [SNELL 2001] SNELL, James. **Programming Web Services with SOAP**. 1ªed. São Paulo: Editora Berkeley, 2001.
- [STALLINGS 1998] STALLINGS, William. **Cryptography and Network Security: Principles and Practice**. 2ª Edição. New Jersey: Prentice Hall, 1998. p.21-44, 63-199, 253-262.

- [TECHGUIDE 2001] RYAN, Jerry. **QoS in the Enterprise**. Techguide.com. Janeiro de 2001. Disponível em: <http://www.techguide.com>. Acessado em: 10 de Setembro de 2004.
- [UML 2003] OMG. **Unified Modeling Language Specification**. Versão 1.5. Março de 2003. Disponível em: <http://www.omg.org/cgi-bin/doc?formal/03-03-01>. Acessado em: 10 de Outubro de 2004.
- [URIS 1998] *RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax*, Berners-Lee T., Fielding R., Masinter L., IETF, Agosto de 1998, <http://www.isi.edu/in-notes/rfc2396.txt>.
- [W3C 1999] W3C. **Namespaces in XML**. 1999. Disponível em: <http://www.w3.org/TR/REC-xml-names/>. Acessado em: 12 de Junho de 2004.
- [W3C 2004] W3C. **RDF Primer**. 2004. Disponível em: <http://www.w3.org/TR/rdf-primer/>. Acessado em: 12 de Junho de 2004.
- [WEBOPIDIA 2004] Webopidia. Disponível em <http://www.webopidia.com>. Acessado em: 5 de Outubro de 2004. Termos: SSL, IPsec, Cryptography, public-key encryption.
- [WEBSOL 2004] WEBSOL. **PiX Insights**. Disponível em: http://www.pixsoft.com.br/web_sol.htm. Acessado em: 10 de Fevereiro de 2004.
- [WILSON 2002] WILSON, Brendon J. **JXTA**. 1ª Ed. Indianapolis – Indiana: New Riders Publishing, 2002.
- [SUNONE 2004] SunOne. **SOAP Clients and Services Using SAAJ and JAXM**. Figura. Disponível em <http://docs.sun.com/source/816-7152-10/wsgjaxm.html>. Acessado em: 28 de Novembro de 2004.

[ZDNet 2002] LONEY, Matt. **The State of Web Services**. ZDNet UK. Disponível em:
<http://www.zdnet.co.uk/i/z/tu/illo/WebServices.gif>. Acessado em: 28 de
Novembro de 2004.

7. ANEXO A

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Modelo de uma Biblioteca Virtual com Peer-to-Peer e Web Services

Fábio Schmitz Tani

Sistemas de Informação

Resumo

Essa pesquisa visa estudar técnicas de segurança para estabelecer confiança no conteúdo de redes *peer-to-peer* públicas ou privadas, sem diminuir a habilidade dos usuários da rede de poderem contribuir para com a rede com conteúdo.

Palavras-chave: P2P, *peer-to-peer*, compartilhamento de arquivos, metadados, redes distribuídas.

Abstract

This research proposes to study security techniques to establish trust in the content of private and public peer-to-peer networks, without diminishing the ability of the users to contribute with the publishing of content in the network.

Palavras-chave: P2P, *peer-to-peer*, file-sharing, metadata, distributed networks.

1. PEER-TO-PEER

O termo *peer-to-peer* apesar de parecer um termo novo, é na verdade baseado na primeira forma de implementação das redes, onde elas eram ligadas ponto a ponto, um nodo ao outro. [ORAM 2001]

Com o crescimento da internet, o ponto-a-ponto foi dando espaço à arquitetura cliente/servidor (Figura 1a), que é a atual estrutura da internet, onde vários provedores de serviços servem vários clientes. Na arquitetura cliente/servidor, os clientes apenas usam os serviços, e nunca colaboram com a rede, muitas vezes

por causa de *gateways*²⁶ e/ou *firewalls*²⁷, que tiram a habilidade deles de se tornarem provedores de serviços.

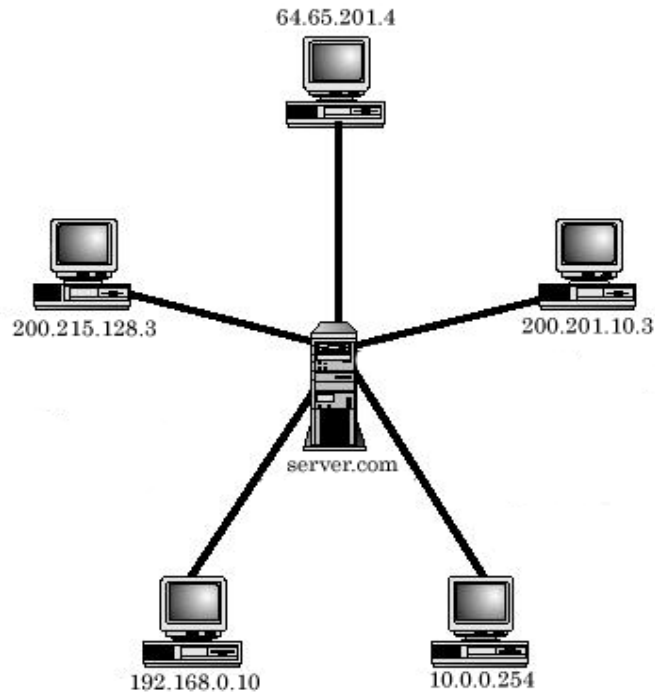


Figura 1a – Arquitetura Cliente/Servidor

A recente arquitetura *peer-to-peer*, nada mais é do que a re-implementação da velha rede ponto-a-ponto adequando-a as redes de hoje em dia, possibilitando aos nodos tornarem-se provedores de serviços, mesmo atrás de *gateways* e *firewalls* (Figura 1b).

²⁶ Nodo da rede que liga uma rede privada com outra rede. Ex.: Gateway de internet liga os nodos da rede local à internet.

²⁷ Nodo da rede ou software que bloqueia tráfego para dentro ou para fora da rede, usado para proteger os recursos da rede contra invasores.

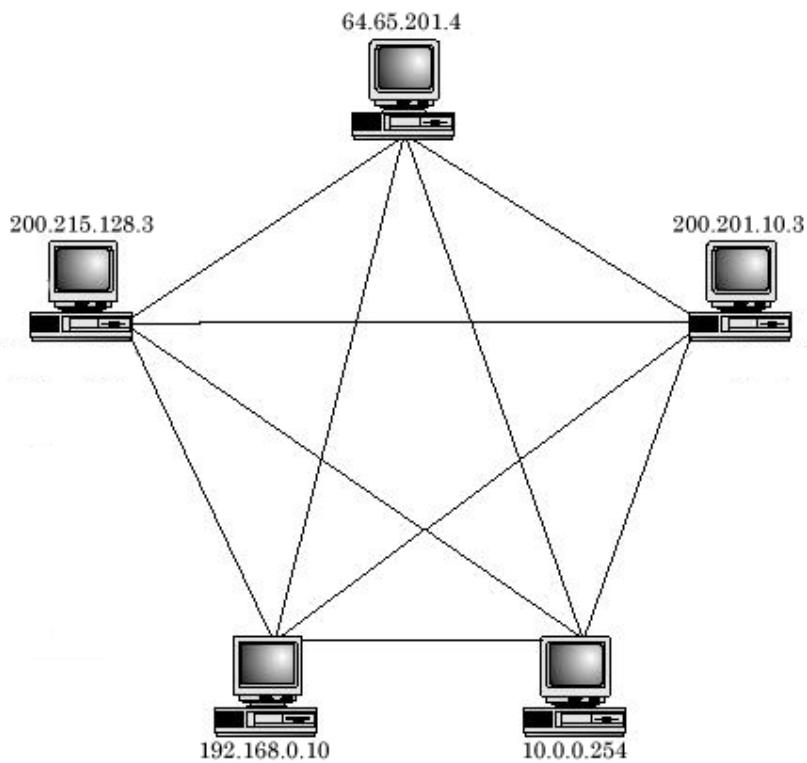


Figura 1b – Arquitetura *peer-to-peer*

1.1 Elementos de uma rede *peer-to-peer*

1.1.1 Peers

Um *peer* é um nodo em uma rede *peer-to-peer*, e é definido como “qualquer entidade capaz de fazer algum trabalho útil e comunicar os resultados desse trabalho para outra entidade dentro da rede, direta ou indiretamente” [WILSON 2002].

Essa definição de trabalho útil depende do tipo do peer, os quais podem ser separados em três tipos:

- **Peers simples:** São usuários simples, normalmente atrás de *firewalls*, por causa do seu acesso limitado eles tem o menor nível de responsabilidade na rede;

- **Peers rendez-vous:** Responsáveis por proporcionar aos outros *peers* da rede informações sobre outros *peers* que eles têm conhecimento. Essas informações são obtidas através de uma *discovery query* feita por um *peer* ao rendez-vous peer;
- **Peers Roteadores:** Fazem a ponte entre os *peers* que estão atrás de *firewalls* ou *gateways*. Agem de forma similar aos DNS, mas de forma dinâmica, e providenciam os *peers* atrás da rede com uma representação que pode ser usada para se comunicar com *peers* fora da rede.

1.1.2 Grupos de *peers*

As aplicações de hoje em dia, pela sua natureza proprietária e especializada, acabaram dividindo o espaço da rede *peer-to-peer* entre as redes de cada aplicação, a exemplo, as redes Gnutella que se comunicam somente com redes Gnutella. Quando a rede *peer-to-peer* é composta de aplicações que usam os mesmos protocolos, como é feito com o JXTA, é necessário o conceito de grupos de *peers* para subdividir o espaço na rede.

Grupos de *peers* não são necessariamente abertos a qualquer nodo da rede. Eles podem ser divididos com base na aplicação utilizada, ou talvez necessitem de um nível de acesso de segurança por parte do *peer*. Alguns grupos, entretanto, requerem que o *peer* forneça algum serviço para poder se conectar ao grupo, ou aproveitar dos recursos do grupo.

1.1.3 Transporte de rede

Para efetuar a troca dos dados, os *peers* usam de algum mecanismo para troca de dados, seja um tipo de protocolo de transporte de baixo nível, a estilo do TCP ou UDP, ou de alto nível, como o HTTP ou o SMTP.

O conceito do transporte de rede pode ser quebrado em três partes, os *endpoints*²⁸, os *pipes*²⁹ e as *messages*³⁰. Para enviar os dados de um *peer* ao outro, o *peer* remetente deve empacotar os dados a serem transmitidos em uma mensagem usando um *pipe* de saída, ligado ao *endpoint* que age como fonte dos dados, no outro extremo outro *peer* recebe a mensagem através de um *pipe* de entrada, ligado ao *endpoint* que age como destinação final, e extrai os dados transmitidos.

1.1.4 Serviços

São as funcionalidades que os *peers* na rede disponibilizam ao outros *peers*, e englobam todas as atividades que a rede de *peer-to-peer* foi projetada para fazer ou disponibilizar. Esses serviços podem ser divididos em duas categorias:

- **Serviços de *peers*:** Uma funcionalidade proporcionada por um dos nodos da rede, disponível somente quando este nodo estiver conectado a rede;
- **Serviços de grupos de *peer*:** Uma funcionalidade oferecida aos membros de um grupo, ela é disponibilizada por vários membros do

²⁸ “A fonte inicial ou destinação final de qualquer pedaço de dado sendo transmitida na rede” [WILSON 2002]

²⁹ “Canais de comunicação unidirecionais, assíncronos e virtuais conectando dois ou mais *endpoints*” [WILSON 2002].

³⁰ “Container para os dados sendo transmitidos sobre um *pipe* de um *endpoint* ao outro” [WILSON 2002]

grupo, provendo acesso redundante ao serviço. Enquanto um nodo estiver conectado, o serviço está disponível na rede.

1.1.5 Propaganda

“Uma representação estruturada de uma entidade, serviço, ou recurso tornado disponível por um *peer* ou grupo de *peers* como parte da rede *peer-to-peer*” [WILSON 2002].

1.1.6 Protocolos

Um protocolo é apenas uma forma de dizer como alguma ação deve ser feita, no caso do *peer-to-peer* eles são necessários para definir os tipos de interação que um *peer* pode fazer como parte da rede.

O uso da propaganda simplifica os protocolos necessários em uma rede *peer-to-peer*, pois elas definem a estrutura e representação dos dados. Os protocolos organizam então a troca de propagandas que contém a informação necessária. [WILSON 2002]

1.1.7 Nome de entidades

A maioria dos componentes de uma rede *peer-to-peer* precisa de um identificador único que as identifique. A rede *peer-to-peer* ideal qualquer dispositivo deve poder participar, independente de sistema operacional ou transporte de rede. [WILSON 2002]

Alguns exemplos de identificação necessários na rede *peer-to-peer*:

- **Peers:** Precisam ser identificados por uma entidade que permita que outros *peers* o localizem na rede;

- **Grupos de peers:** Um *peer* precisa saber qual o grupo em que ela fará alguma ação;
- **Pipes:** A comunicação necessita que seja identificado, qual *pipe* está conectado a qual *endpoint*;
- **Conteúdos:** O conteúdo da rede deve possuir um identificador único, que possibilite que os *peers* o identifiquem pela rede, e que haja redundância da informação entre os *peers*.

1.1.8 Projeto JXTA

É um *framework*³¹ para aplicações *peer-to-peer*, foi iniciado como um projeto da Sun Microsystems, com o objetivo de explorar a visão da computação distribuída usando uma topologia *peer-to-peer*. O projeto continua em andamento, mas seu código é aberto, e conta com a ajuda de muitos membros da comunidade Java. Ele fornece toda a infra-estrutura para criar redes *peer-to-peer*, usando nodos, super-nodos³², grupos e outras tecnologias do *peer-to-peer*.

Por ser um projeto em andamento, a cada dia surgem novas mudanças e alterações ao seu código, entretanto o desenvolvimento dessa ferramenta já está em um estado estável, que possibilita a criação de plataformas inteiras em cima do *framework*.

³¹ Wirfs-Brock (apud Silva, 2000) “Um esqueleto de implementação de uma aplicação de um subsistema de aplicação, em um domínio de problema particular. É composto de classes abstratas e concretas e provê um modelo de interação ou colaboração entre as instâncias de classes definidas pelo framework. Um framework é utilizado através de configuração ou conexão de classes concretas e derivação de novas classes concretas a partir das classes abstratas do framework”.

³² Nodos da rede que atuam como servidores com a finalidade de localizar os clientes conectados a rede P2P.

2. MODELO DE BIBLIOTECA VIRTUAL

A biblioteca virtual pode ser considerada um serviço na web, pois provê aos usuários o acesso a um conteúdo através do acesso remoto e por ser considerado um serviço web, deve seguir alguns conceitos para ser bem sucedido.

2.1 Segurança

Uma parte fundamental de quaisquer serviços na web é a segurança, esta pode assumir várias formas. Em especial para sistemas distribuídos, como é o caso do *peer-to-peer*, necessitamos de quatro formas de segurança básicas:

- Confidencialidade: proteger a informação trocada de acessos não autorizados;
- Integridade: apenas os autorizados podem modificar a informação;
- Autenticidade: identificar que a origem e o destino da mensagem não são falsos;
- Disponibilidade: os serviços devem estar sempre disponíveis a uma margem de usuários.

2.2 Segurança para a biblioteca virtual

O modelo de uma fonte de conhecimento segura, deve abranger mais do que as cinco formas de seguranças vistas. Além de prover a segurança dos dados trafegados e de quem está navegando na rede, e necessário prover também a segurança do conteúdo da rede.

Por causa da natureza do software se faz necessária que algumas medidas de segurança sejam aplicadas em cima dos nodos que compartilham os dados da rede, estes métodos de segurança variam com o estilo da rede.

Quando a rede é denominada privada e o conteúdo da rede é restrito a uma ou mais instituições que desejam que esse conteúdo seja apenas aquele publicado pelas suas autoridades delegadas, é utilizado um modelo de nodo autoridade que restringe o conteúdo da rede, visto na seção 2.2.1.

Quando a rede é denominada pública, mas ainda é necessário que o conteúdo publicado pelas autoridades da rede seja diferenciado, é utilizado um modelo de nodos credenciadores, que identificam conteúdo publicado pelas autoridades delegadas, visto na seção 2.2.2.

2.2.1 Nodos autoridades

Como o conteúdo da rede são documentos autênticos, é necessário que a publicação de arquivos na rede seja permitida para apenas documentos autênticos legítimos. Para isso é utilizada a idéia de autoridades dentro da rede, aonde somente essas autoridades podem publicar novos materiais. A figura 2.2.1a ilustra uma rede *peer-to-peer* com um nodo autoridade:

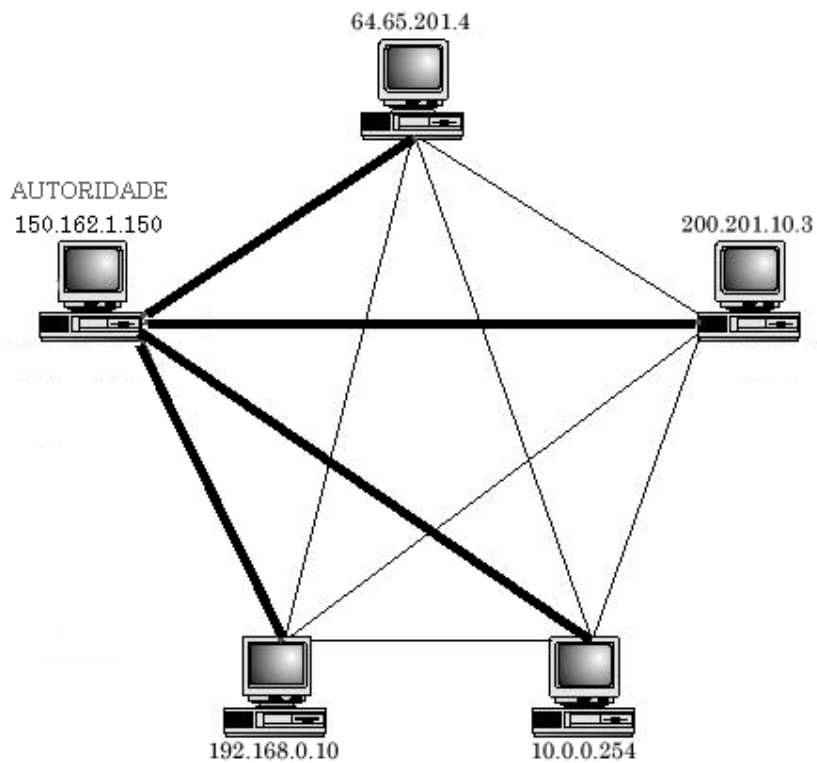


Figura 2.2.1a – Rede *peer-to-peer* com nó autoridade

Nodos autoridade, possuem a função de publicar conteúdo novo na rede, e autenticar a publicação de conteúdo por nodos normais. Os nodos normais têm permissão para publicar somente conteúdo já publicado pelos nodos autoridade e que estão presentes na rede, assim garantindo que os dados da rede são legítimos e confiáveis.

Os nodos autoridades, na concepção da biblioteca virtual, devem ser constituídos somente das instituições responsáveis pelo conteúdo da rede.

A autenticação de conteúdo é dada através do armazenamento de resumos (hash) dos arquivos, e de uma criptografia de chave pública para trafegar as mensagens de pedido de publicação de forma segura. A figura 2..2.1b ilustra o processo de publicação de conteúdo de um nodo normal.

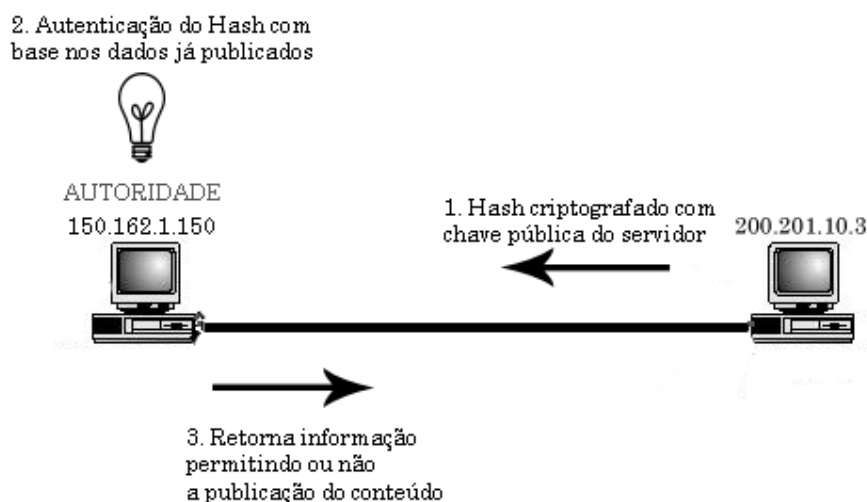


Figura 2.2.1b – Esquema de publicação

2.2.2 Nodos credenciadores

Quando o objetivo da rede é ser totalmente pública, ou seja permitir que os usuários contribuam com a rede com quaisquer tipos de documentos, a fim de enriquecer a rede, é utilizada uma abordagem similar a vista na seção 2.2.1.

Nodos credenciadores são baseados nos nodos autoridades, entretanto não exercem a função de permitir ou não a publicação de conteúdo na rede. Os nodos credenciadores têm como função básica diferenciar o conteúdo credenciado pelas autoridades, e o conteúdo publicado pelos usuários da rede.

Outras funções de um nodo credenciador são a publicação de conteúdo na rede e o credenciamento de conteúdo da rede. O credenciamento de conteúdo pode ser feito a qualquer conteúdo publicado na rede, mesmo que esse conteúdo não tenha sido publicado por um nodo credenciador, desde que o conteúdo já não esteja credenciado.

Essa diferenciação de conteúdo garante que os documentos credenciados pelos nodos credenciadores sejam assegurados, e mesmo assim permite que a rede

suporte a adição de novos documentos pelos usuários da rede, a fim de enriquecer o conteúdo da rede com a contribuição dos usuários.

3. ANÁLISE DO MODELO DE BIBLIOTECA VIRTUAL PARA UMA BIBLIOTECA VIRTUAL PÚBLICA

A biblioteca é um sistema distribuído, que contém dois tipos de componentes, os softwares para usuários comuns da biblioteca, e os nodos credenciadores.

O usuário comum dessa biblioteca pública tem como objetivo base pesquisar informação na rede, e fazer o download dessa informação. Por se tratar de uma rede *peer-to-peer*, o usuário pode contribuir para com a rede, através da publicação de documentos novos ou já contidos na rede, de forma a criar uma redundância e balanceamento de carga da informação compartilhada na rede.

O nodo credenciador, por sua vez, tem como objetivo base, fornecer conteúdo para a rede, e assegurar que o conteúdo da rede marcado como credenciado é seguro e confiável. O nodo credenciador deve assegurar que os documentos credenciados não tenham seu conteúdo alterado.

3.1 Modelo Conceitual

Utilizando como referência o framework JXTA, é proposto um modelo conceitual de implementação das principais classes para a biblioteca virtual (Figura 3.1a), esse modelo mostra a classe pai *Nodo* que guarda os atributos e métodos comuns a todos os tipos de nodos, também é mostrado o framework JXTA, sendo que muitos tipos utilizados nso modelo são provindos desse framework.

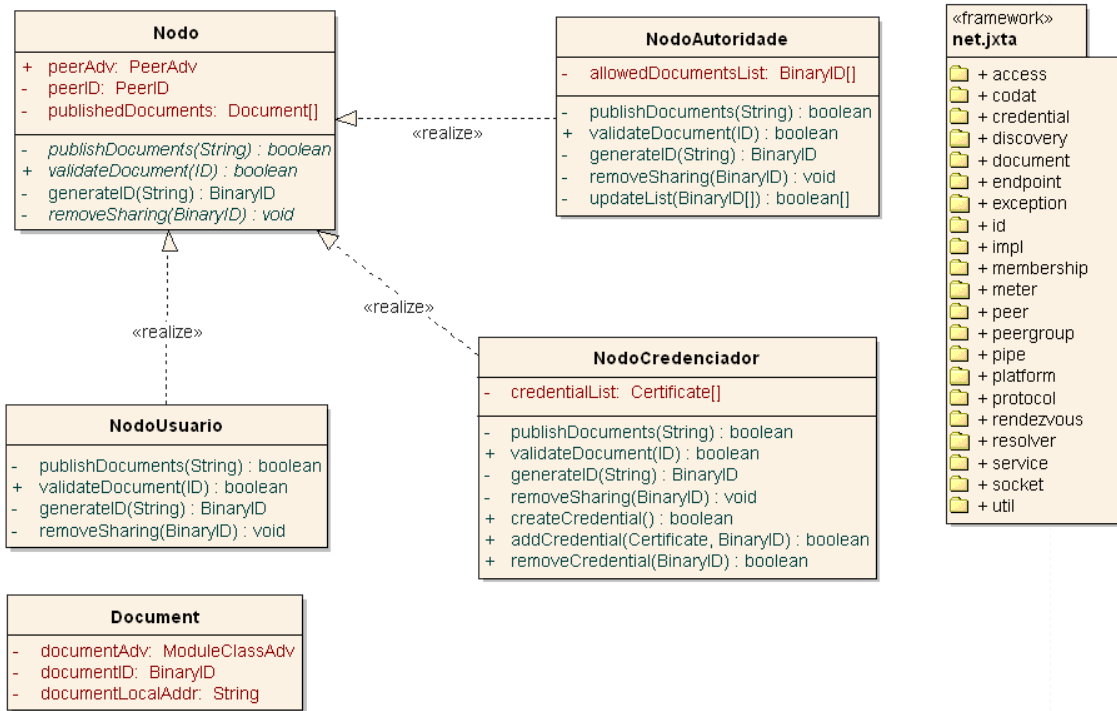


Figura 3.1a – Modelo Conceitual da Biblioteca Virtual

4. REFERÊNCIAS BIBLIOGRÁFICAS

[ORAM 2001] ORAM, Andy. **PEER-TO-PEER: O poder transformador das redes ponto a ponto**. São Paulo: Editora Berkeley, 2001.

[SILVA 2000] SILVA, R. P. **Suporte ao desenvolvimento e uso de frameworks e componentes**. Tese de doutorado. Porto Alegre: UFRGS/II/PPGC, mar. 2000. 262p. Disponível em: <http://www.inf.ufsc.br/~ricardo/download/tese.pdf>. Acesso em: 12 de junho de 2004.

[WILSON 2002] WILSON, Brendon J. **JXTA**. 1ª Ed. Indianapolis – Indiana: New Riders Publishing, 2002.