

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Andréa Sabedra Bordin

**Extensão do Protocolo Z39.50 para Busca de
Servidores de Bibliotecas Digitais**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos
requisitos para a obtenção do grau de Mestre em Ciência da Computação

Prof. Roberto Willrich

Florianópolis, 24 de outubro de 2002.

Extensão do Protocolo Z39.50 para Busca de Servidores de Bibliotecas Digitais

Andréa Sabedra Bordin

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Dr. Fernando A. O. Gauthier
Coordenador do Curso

Banca Examinadora

Prof. Dr. Roberto Willrich (Orientador)
INE/UFSC

Prof. Dr. Vitório Bruno Mazzola
INE/UFSC

Prof. Dr. Rosvelter Coelho da Costa
INE/UFSC

Agradecimentos

Acima de tudo agradeço a Deus. Logo após, a todas as pessoas que me apoiaram e incentivam na difícil decisão de “largar tudo” e vir para Florianópolis concluir o Mestrado. Aos amigos que me apoiaram nos momentos difíceis vividos aqui. E de maneira muito especial ao professor Roberto Willrich, orientador dessa dissertação, que verdadeiramente cumpriu a sua função orientando com muita seriedade e competência esse trabalho.

Sumário

Capítulo 1	Introdução.....	1
1.1	Objetivo.....	2
1.2	Estrutura do Documento.....	3
Capítulo 2	Bibliotecas Digitais.....	4
2.1	Conceito e Classificação.....	4
2.2	Características de uma Biblioteca Digital	5
2.3	Biblioteca Digital x (Biblioteca Tradicional e Web)	5
2.4	Etapas para o desenvolvimento de Bibliotecas Digitais.....	7
2.4.1	Criação e captura	7
2.4.2	Gerência e armazenamento.....	7
2.4.3	Busca e acesso.....	8
2.4.4	Disponibilização	8
2.4.5	Tratamento de direitos autorais	8
2.5	O Papel dos Metadados no Contexto da Biblioteca Digital	8
2.5.1	MARC (Machine Readable Cataloging)	9
2.6	Tipos de Arquitetura de Biblioteca Digital	10
2.7	Interoperabilidade em Bibliotecas Digitais	12
2.8	Iniciativas Nacionais e Internacionais de Construção de Bibliotecas Digitais...	13
2.9	Conclusão	15
Capítulo 3	Protocolo Z39.50.....	13
3.1	Histórico de Desenvolvimento.....	13
3.2	Modelo de Funcionamento do Z39.50.....	14
3.3	Provendo a Interoperabilidade	16
3.3.1	Mecanismos de Codificação	17
3.3.2	Conteúdo Semântico	20
3.4	Definição de Serviços do Protocolo	20
3.5	Especificação da Transferência de Informação no Protocolo	23
3.6	Conclusão	23
Capítulo 4	Extensão do Z39.50.....	24
4.1	Descrição e Avaliação das Alternativas	25
4.1.1	Busca Centralizada	25
4.1.2	Busca Distribuída a partir do Servidor.....	26
4.1.3	Busca a partir do Cliente	33
4.1.4	Análise das Alternativas	34
4.2	Detalhamento da Busca Distribuída a partir do Cliente, com Organização Hierárquica.....	36
4.2.1	Descrição do Serviço <i>Searchlib</i> –Serviço de Busca de Servidores de Bibliotecas Digitais	36
4.2.2	Variáveis de Estado do Servidor.....	38
4.2.3	Tabela de Nomes	39
4.2.4	Hierarquia de Servidores	40

4.2.5	Organização da Hierarquia de Servidores	49
4.3	Cenário de Uso	50
4.3.1	Cliente e Servidor.....	50
4.3.2	Processo de Busca de Bibliotecas Digitais	50
4.4	Validação da Proposta.....	56
4.5	Conclusão	59
<i>Capítulo 5</i>	<i>Conclusão Final.....</i>	<i>60</i>
	<i>Referências Bibliográficas</i>	<i>62</i>
	<i>Anexo 1 – Descrição dos Serviços do Protocolo Z39.50.....</i>	<i>65</i>
	<i>Anexo 2 – Código Fonte da Simulação de Funcionamento do Serviço Searchlib.....</i>	<i>89</i>

Lista de Figuras

Figura 1. Arquitetura Distribuída de Biblioteca Digital	11
Figura 2. Comunicação entre Cliente e Servidor através do Protocolo Z39.50	17
Figura 3. Itinerário de Busca de Informação através da Utilização do Protocolo Z39.50	19
Figura 4. Implementação do Z39.50 da Perspectiva do Usuário	21
Figura 5. Implementação da Perspectiva do Z39.50	22
Figura 6. Organização Encadeada de Servidores de Bibliotecas Digitais	30
Figura 7. Inclusão de um Novo Servidor na Organização Encadeada	31
Figura 8. Formato da Tabela de Nomes de Servidor de Biblioteca Digital	41
Figura 9. Escopo Local de uma Organização de Servidores de Bibliotecas Digitais	43
Figura 10. Tabela de Nomes de um Servidor de Nomes Local-Biblioteca Universitária	43
Figura 11. Tabela de Nomes de um Servidor Local-Biblioteca do CCS	44
Figura 12. Escopo Regional de uma Organização de Servidores de Bibliotecas Digitais	46
Figura 13. Tabela de Nomes de um Servidor de Nomes Regional-Biblioteca Universitária	47
Figura 14. Escopo Nacional de uma Organização de Servidores de Bibliotecas Digitais	49
Figura 15. Tabela de Nomes de um Servidor de Nomes Nacional-Biblioteca Universitária da UFSC	50
Figura 16. Tabela de Nomes de um Servidor de Nomes Global-Biblioteca Universitária da UFSC	51
Figura 17. Interface para a Escolha do Escopo de Busca de Bibliotecas Digitais	54
Figura 18. Interface para a Entrada de Dados Necessários para a Pesquisa de Documento Digitais	54
Figura 19. Interface para a Escolha do Servidor Local Alvo da Pesquisa	56
Figura 20. Interface para a Entrada de Dados Necessários para a Pesquisa de Documentos Digitais	57
Figura 21. Interface com o Resultado da Ramificação de uma Busca de Bibliotecas Digitais	58
Figura 22. Cenário dos Servidores de Biblioteca Digitais Utilizados na Simulação	59

Lista de Tabelas

Tabela 1. Desvantagens das Alternativas de Busca de Bibliotecas Digitais	38
Tabela 2. Parâmetros do Serviço <i>Searchlib</i>	43
Tabela 3. Valores Possíveis do Campo <i>Scope</i> do Serviço <i>Searchlib</i>	40
Tabela 4. Formato do Campo <i>ResultSet</i> do Serviço <i>Searchlib</i>	40
Tabela 5. Mapeamento dos Pedidos de <i>Searchlib</i> aos Servidores de Bibliotecas Digitais Utilizados na Simulação	59

Resumo

O Z39.50 é um protocolo de aplicação cliente/servidor utilizado para a busca e recuperação de informação distribuída. Esse protocolo é muito utilizado no contexto das bibliotecas digitais, pois propicia que um cliente (usuário) estabeleça comunicação com um servidor de biblioteca digital e efetue a busca e recuperação de documentos digitais. Para que o cliente se comunique com o servidor é necessário que a aplicação do cliente conheça previamente, ou seja, em tempo de programação, o endereço do servidor com o qual a comunicação será estabelecida. Entende-se que isso se concretiza numa limitação na busca e recuperação de documentos digitais já que atualmente existem muitas bibliotecas digitais e a tendência é que esse número aumente. A situação ideal é que informações como nome e endereço de servidores de bibliotecas digitais apareçam de forma dinâmica na aplicação do cliente para que então o usuário possa escolher em qual ou em quais servidores deseja efetuar a pesquisa por documentos digitais. Essa dissertação objetiva sanar essa limitação propondo uma extensão no protocolo Z39.50 no sentido da criação de um novo serviço chamado *Searchlib* que tem a função de buscar e recuperar para o cliente informações (nome e endereço) de servidores de bibliotecas digitais, permitindo com isso que eles apareçam de forma dinâmica na aplicação do cliente.

Abstract

Z39.50 is a protocol of application client/server used for the search and recovery of distributed information. That protocol is very used in the context of the digital libraries, because it propitiates that a cliente (user) it establishes communication with a server of digital library and make the search and recovery of digital documents. So that the client communicates with the server is necessary that the client's application previously knows, in other words, in time of programming, the address of the server with which the communication will be established. We understand each other that is rendered in a limitation in the search and recovery of digital documents since now they exist many digital libraries and the tendency is to increase that number. The ideal situation is that information as name and addresses of servers of digital libraries appear in a dynamic way in the customer's application so that then the user can choose in which or in which servers want to make the research for digital documents. That dissertation aims at to heal that limitation proposing an extension in the protocol Z39.50 in the sense of the creation of a new called service Searchlib that has the function of to search for and to recover for the client information (name and address) of servers of digital libraries, allowing with that that they appear in a dynamic way in the client's application.

Capítulo 1 Introdução

Durante um certo período da história da humanidade, mais especificamente durante a idade média, as bibliotecas foram consideradas fontes muito veladas de conhecimento, onde o acesso era muito restrito, ou seja, somente os pensadores e intelectuais da época tinham acesso a essas fontes de conhecimento. Os séculos foram se passando e acesso às bibliotecas se tornou muito mais fácil, contribuindo para que o objetivo principal de uma biblioteca, que é de disseminar conhecimento, se ampliasse bastante.

As bibliotecas começaram a ampliar seus acervos, até então estritamente físicos e buscar uma informação começou a se tornar uma tarefa morosa e complicada. Surgiram os primeiros indícios de catalogação e a procura por obras através de fichas, o que resolveu em parte o problema do acesso rápido à informação.

Em meados do século passado, a tecnologia computacional começou a crescer vertiginosamente, e os primeiros sistemas para gerenciamento de bibliotecas foram criados. Esses sistemas ainda estão presentes na maioria das bibliotecas e permitem o cadastramento do acervo, a emissão de etiqueta de catalogação, e em muitos casos gerenciam o empréstimo e devolução das obras e permitem a pesquisa de localização de determinada obra dentro do acervo físico da biblioteca.

O advento da Internet através da *World Wide Web* propiciou uma grande distribuição de informações, o acesso à informação deixou de ficar refinado a um local físico para se tornar acessível a qualquer pessoa em qualquer local, permitindo a disponibilização de informações até então em mídia impressa em um formato digital.

Se beneficiando dos conceitos que a Internet introduziu, como a distribuição, acesso rápido e disponibilização no formato digital de informações surge o conceito de biblioteca digital que segundo [GLADNEY, 1994] “é a reunião de um ferramental de computação, estoque e comunicação digitais juntamente com o conteúdo e o software necessário para se reproduzir, emular, e estender os serviços oferecidos por bibliotecas convencionais baseadas em papel e outros meios de coleção, catalogação, e disseminação da informação. Uma biblioteca digital completa deve ser capaz de

oferecer todos os serviços essenciais de uma biblioteca tradicional, assim como explorar as bem conhecidas vantagens do estoque, pesquisa e comunicação digitais.”

A característica principal que acompanha o conceito de biblioteca digital e que também a difere da WWW, é a de disponibilizar o acesso e busca a documentos de qualidade à qualquer pessoa e de qualquer lugar. Acessar documentos de qualidade significa tornar disponível no formato digital, através de digitalização ou outro meio, documentos até então somente disponíveis fisicamente (em papel) em determinada biblioteca tradicional.

Para que a questão do acesso e busca por qualquer pessoa em qualquer lugar a documentos de uma biblioteca digital se torne mais eficiente é necessário tornar as bibliotecas digitais interoperáveis, ou seja, operáveis independentemente de sua forma de armazenar os documentos digitais. Para isso, é de suma importância o desenvolvimento de ferramentas independentes de acesso e busca a informações disponibilizadas em qualquer biblioteca digital interoperável. Portanto, é fundamental a existência de protocolos abertos que se encarreguem de estabelecer a comunicação entre as bibliotecas digitais (servidores) e os usuários (clientes) e propiciem o acesso e a busca de informação. Entre esses protocolos destaca-se o Z39.50 [Z39.50, 2002], um protocolo criado para a busca e recuperação de informação distribuída.

O Z39.50 é um protocolo de aplicação cliente/servidor, ou seja, ele estabelece uma comunicação entre uma máquina que implementa o módulo cliente do protocolo e uma máquina que implementa o módulo servidor do protocolo.

1.1 Objetivo

Como o protocolo Z39.50 se encontra atualmente, é necessário que a aplicação do módulo cliente do protocolo tenha conhecimento em tempo de desenvolvimento de informações como o nome e endereço de servidores de bibliotecas digitais onde a busca e recuperação da informação será efetuada. Entende-se que essa situação não é a mais apropriada, pois limita a busca apenas às bibliotecas digitais cujos endereços são conhecidos previamente e se encontram disponibilizados na aplicação. Para sanar essa limitação, o ideal é que seja criada na aplicação do cliente uma lista com nomes e

endereços de servidores de biblioteca digitais e que a criação dessa lista aconteça de forma dinâmica, ou seja, no momento em que o usuário necessita realizar alguma pesquisa. A criação da lista com informações como nome e endereço de servidores de bibliotecas digitais pode ser caracterizada como um “Portal Dinâmico”, pois tem a função de permitir o acesso a partir de uma única interface a várias bibliotecas digitais e deve ser construída dinamicamente.

Para concretizar a situação ideal apresentada acima, está se propondo uma extensão no protocolo Z39.50 no sentido da criação de um novo serviço que se encarrega de buscar informações como nome e endereço de servidores de bibliotecas digitais que implementam o protocolo Z39.50 e disponibilizar essas informações de forma dinâmica para a aplicação do cliente. A idéia de extensão permite que o protocolo fique como está originalmente em seus outros serviços, apenas se propõe um novo serviço que é executado primeiramente no caso do usuário querer efetuar a busca por bibliotecas digitais.

Neste caso, são emitidos pedidos de busca de servidores de bibliotecas digitais para servidores de bibliotecas digitais e é esperado como resposta informações como o nome e endereço de outros servidores de bibliotecas digitais, permitindo com isso que seja construído dinamicamente na aplicação do cliente uma lista com informações (nome e endereço) de servidores de bibliotecas digitais, através da qual o usuário poderá escolher em qual ou em quais servidores realizará a pesquisa por documentos digitais.

1.2 Estrutura do Documento

Os próximos capítulos abordam conceitos pertinentes a essa proposta.

O capítulo 2 trata dos conceitos relacionados à bibliotecas digitais, desde o conceito básico de biblioteca, suas classificações, os tópicos relacionados à construção e utilização de bibliotecas digitais até as iniciativas nacionais e internacionais de construção de bibliotecas digitais.

O capítulo 3 aborda o protocolo Z39.50, desde o histórico de desenvolvimento, modelo de funcionamento, definição de serviços de aplicação à especificação do protocolo em si.

O capítulo 4 apresenta a proposta de extensão do protocolo Z39.50 através da criação de um novo serviço, o qual possibilitará a busca de servidores de bibliotecas digitais à serem pesquisados na aplicação do cliente. Neste capítulo é apresentada uma análise das alternativas estudadas para a implementação desse serviço, bem como o formato das primitivas de pedido e resposta do serviço, além de outras variáveis envolvidas.

Por fim, o capítulo 5 apresenta as conclusões deste estudo que se efetivaram através de uma aplicação desenvolvida com o objetivo de avaliar o funcionamento da proposta e as perspectivas futuras deste trabalho.

Capítulo 2 Bibliotecas Digitais

Este capítulo introduz o conceito genérico de biblioteca, suas classificações e se aprofunda nos aspectos relacionados a um tipo específico de biblioteca, a biblioteca digital.

2.1 Conceito e Classificação

O termo biblioteca pressupõe a existência de um local físico em uma determinada posição geográfica que armazena coleções de livros, periódicos, etc. Essa é a chamada biblioteca convencional ou tradicional, que existe desde há muito tempo e que hoje se encontra na maioria dos locais (cidades, instituições escolares, etc.).

A biblioteca convencional passou por evoluções e hoje, segundo [BARKER, 1994], ela pode ser chamada de polimídia, co-existindo com mais três tipos de bibliotecas: eletrônica, digital e virtual. A biblioteca polimídia se caracteriza por possuir um acervo físico composto de livros, periódicos, e de conviver com fitas de vídeo, CD-ROMs, microfimes, etc. O acesso ao acervo é realizado através de referências bibliográficas disponibilizadas em fichas e arquivos.

A biblioteca eletrônica consiste de materiais eletrônicos e serviços para facilitar a pesquisa e a consulta ao acervo. Materiais eletrônicos podem incluir não só materiais digitais, como também uma variedade de formatos analógicos (equipamentos de microfilmagem e fitas de vídeo, por exemplo) que exigem dispositivos eletrônicos para seu uso. [TENNANT,1999]

A biblioteca digital contém apenas informação na forma digital, ou seja, o acervo se encontra disponibilizado no formato digital, podendo residir em meios diferentes de armazenamento.

A biblioteca virtual é conceituada como um tipo de biblioteca que, para existir, depende da tecnologia da realidade virtual, ou seja, oferece aos usuários efeitos de realidade virtual, disponibilizando uma forma de utilização similar à de uma biblioteca tradicional. [PEREIRA, 1995]

2.2 Características de uma Biblioteca Digital

A biblioteca digital se difere das demais, à medida que suas informações existem somente na forma digital, não contendo livros, fitas e CD-ROMs na forma convencional. Todos os itens da biblioteca digital estão armazenados no formato digital, oferecendo pesquisa e visualização dos mesmos tanto local, como remotamente.

Ela deve permitir o acesso e armazenamento de vários tipos de mídias, estáticas (imagens, gráficos) e dinâmicas (áudios e vídeos), podendo nesse caso, ser considerada uma biblioteca digital multimídia.

A coleção de documentos de uma biblioteca digital deve estar disponível para qualquer pessoa que quiser acessá-la (em alguns casos é necessário que o usuário esteja autorizado), em qualquer lugar, o que pressupõe a existência de uma infra-estrutura Internet, ou seja, os acessos à biblioteca digital devem ocorrer geralmente via Internet através do WWW.

2.3 Biblioteca Digital x (Biblioteca Tradicional e Web)

Bibliotecas digitais possuem muitas vantagens em relação às bibliotecas tradicionais:

- ? Acesso fácil e rápido à informação: o acesso ao acervo pode ser realizado de qualquer computador que tenha acesso à Internet, não havendo necessidade de deslocamento físico. A rapidez se verifica na medida em que não há gasto de tempo na busca da informação, o usuário se limita a entrar com algumas informações e os mecanismos de busca se encarregam de encontrar os links para o documento digital e disponibilizá-lo para o usuário
- ? Redução dos custos de aquisição: com a existência de várias bibliotecas digitais, estas podem compartilhar recursos, o que possibilita uma redução no custo de aquisição de obras. Isso permite que bibliotecas se especializem em determinadas áreas adquirindo determinados tipos de obras.

- ? Preservação de documentos: como o acervo está disponibilizado na forma digital, o manuseio físico não existe, possibilitando a preservação do mesmo.
- ? Acesso independente de localização geográfica: o acesso à fonte de informação se torna ilimitado, eliminando-se as barreiras impostas ao acesso da informação pela localização geográfica de algumas comunidades. Qualquer pessoa de qualquer local com acesso a Internet pode dispor da informação. O resultado dessa característica, talvez a mais importante é a diminuição das diferenças sociais, econômicas e políticas.
- ? Acesso por parte de portadores de deficiência: as barreiras impostas a pessoas com alguma deficiência física também caem. Além da facilidade de acesso para as pessoas paraplégicas ou tetraplégicas, alguns recursos no formato digital podem ser transformados por um sintetizador de voz, para o caso dos deficientes visuais.
- ? Facilidade de colaboração educacional e aprendizagem: a utilização de bibliotecas digitais permite que a informação possa circular mais rapidamente, permitindo que docentes e discentes participem do processo de colaboração educacional. O processo de aprendizagem é facilitado pela facilidade na reprodução e acesso à informação.
- ? Variedade de formato de documentos: bibliotecas digitais se caracterizam por armazenar e disponibilizar diferentes formatos de mídias: texto, áudio, vídeo. O acesso à informação de diferentes maneiras possibilita um incremento no conhecimento.

A Web, assim como as bibliotecas digitais, também é considerada um grande repositório de informações e isso faz com que dúvidas apareçam na justificativa de construção de bibliotecas digitais.

Porém, bibliotecas digitais também possuem vantagens em relação à Web. Entre elas pode-se citar:

- ? Precisão dos resultados de pesquisa: devido aos algoritmos de busca dos mecanismos de pesquisa atuais da Web, muita informação fora do contexto é retornada para o usuário. Com o uso de bibliotecas digitais a precisão de retorno de informação é muito maior, pois qualquer documento do acervo digitalizado possui descritores (metadados) associados que auxiliam na precisão da descoberta do recurso.
- ? Acesso a material de qualidade: no ambiente da Web, muita informação de qualidade (fonte) duvidosa é retornada. Os documentos de uma biblioteca digital são selecionados e digitalizados por uma equipe de profissionais especializados.

2.4 Etapas para o desenvolvimento de Bibliotecas Digitais

Segundo [PISTORI, 2000] o desenvolvimento de uma biblioteca digital pressupõe algumas etapas: criação e captura, gerenciamento e armazenamento, busca e acesso, disponibilização e tratamento de direitos autorais.

2.4.1 Criação e captura

Nessa etapa primeiramente são definidos os objetos a serem disponibilizados no formato digital. Estes objetos podem ser produzidos originalmente sob forma digital (documentos produzidos por editores de texto, por exemplo), ou passarem por um processo de digitalização (por exemplo, um mapa). Portanto, a criação envolve a disponibilização de um documento sob forma digital; e a captura envolve a transformação de um documento do formato não-digital para o digital.

2.4.2 Gerência e armazenamento

O armazenamento de um objeto no formato digital implica em necessidade de espaço para armazená-lo. Portanto é de suma importância a definição de mecanismos de armazenamento que, por razões de desempenho, devem prever a distribuição dos objetos em múltiplos servidores e o mais próximo possível dos usuários. Além disto, é necessário definir procedimentos de backup automático e prover recursos de migração para novas tecnologias.

2.4.3 Busca e acesso

A indexação de objetos no formato digital normalmente é feita utilizando-se bases de dados separadas para os índices (metadados) e para os objetos digitais. Estes índices, além de permitir a pesquisa por elementos tradicionais de identificação dos objetos digitais, tais como autor, título, assunto, abstract e palavras-chave, devem permitir também, pesquisa no conteúdo dos objetos digitais, como por exemplo, no texto completo (full text), no conteúdo das imagens (cor, forma, textura, etc). Deve-se definir se a biblioteca digital conterà somente links para o seu acervo, ou se conterà também índices para dados virtuais em outras bibliotecas digitais. As ferramentas de consulta a base de dados devem prever a utilização da lógica booleana, pesquisa em linguagem natural, parâmetros fonéticos e técnicas de inteligência artificial.

2.4.4 Disponibilização

Esta função trata do planejamento da infra-estrutura física de comunicação necessária para que as bibliotecas digitais possam prover acesso a todos os seus objetos digitalizados, por qualquer pessoa, a qualquer hora e de qualquer lugar.

2.4.5 Tratamento de direitos autorais

Estabelece mecanismos de proteção dos documentos contidos em bibliotecas digitais, prevendo critérios para acesso integral ou parcial a objetos digitalizados, mecanismos para liberação de cópias, remuneração dos autores, etc. Atualmente, este é um dos temas que envolvem grande discussão.

2.5 O Papel dos Metadados no Contexto da Biblioteca Digital

Como visto no tópico anterior, duas importantes funções no processo de construção de uma biblioteca digital são a busca e o acesso aos objetos digitais. Isso é implementado através do estabelecimento para cada objeto de um conjunto de descritores. Estes descritores também são conhecidos como *metadados* e se caracterizam por armazenar informações sobre o objeto.

Segundo [SUMPTER, 1994], “Metadado é a informação sobre o dado que permite o acesso e gerenciamento deste dado de maneira eficiente e inteligente.” O conceito de metadado é um conceito antigo e é empregado em diferentes áreas com um

objetivo similar: *permitir uma melhor integração, troca, acesso e interpretação dos dados* [GUNTHER, 1997]. Portanto, metadados podem ser aplicados a uma grande variedade de acervos de dados para descrevê-los, podendo ou não, os dados estarem disponíveis em ambiente eletrônico. Catálogos com informações bibliográficas das bibliotecas tradicionais são exemplos do emprego de metadados, pois nele existem várias informações que descrevem os recursos armazenados, auxiliando os usuários na recuperação da informação.

Não existe um único padrão de metadados, eles são específicos para determinadas áreas de conhecimento. À nível de descrição de informações bibliográficas completas existem os seguintes padrões de metadados: MARC, TEI, EAD e GILS.

Entre esses padrões se destaca MARC - Machine Readable Cataloging, pois este é um padrão de metadados que é utilizado já há algum tempo para descrição de informações bibliográficas.

2.5.1 MARC (Machine Readable Cataloging)

MARC, cuja especificação completa pode ser encontrada em [MARC, 2002], atualmente se denomina MARC 21. Ele foi elaborado para a descrição do acervo físico da Biblioteca do Congresso Americano na década de 60, com o objetivo de promover o intercâmbio de registros de catalogação entre bibliotecas, sendo considerado um padrão entre os catalogadores de bibliotecas americanas.

MARC é composto por um poderoso conjunto de descritores que são explicitados no formato de um registro que pode ser lido por computador (tal qual o nome do padrão). Esse registro é formado por campos (descritores), onde cada campo é identificado por um conjunto de 3 dígitos.

A seguir um exemplo de registro MARC:

100	1#	\$a	Arnosky, Jim.
245	10	\$a	Raccoons and ripe corn /
		\$c	Jim Arnosky.
250	##	\$a	1st ed.
260	##	\$a	New York :
		\$b	Lothrop, Lee & Shepard Books,
		\$c	c1987.
300	##	\$a	25 p. :
		\$b	col. ill. ;
		\$c	26 cm.
520	##	\$a	Hungry raccoons feast at night in a field of ripe corn.
650	#1	\$a	Raccoons.
900	##	\$a	599.74 ARN
901	##	\$a	8009
903	##	\$a	\$15.00

2.6 Tipos de Arquitetura de Biblioteca Digital

Em relação à arquitetura de construção, segundo [PISTORI, 2000] as bibliotecas digitais podem ser centralizadas e distribuídas. A distribuição, aqui, se refere à existência de diversas bibliotecas digitais que poderiam ser acessadas via uma interface única do cliente.

Algumas características das duas arquiteturas são apresentadas abaixo:

- a) **Arquitetura centralizada:** os documentos digitais que compõem as coleções podem ser armazenados em servidores distintos, mas o gerenciamento e a busca são centralizados em um único servidor. Este tipo de arquitetura não leva em consideração a interoperabilidade entre bibliotecas digitais.
- b) **Arquitetura distribuída:** neste caso, o gerenciamento do acervo, a busca e o armazenamento são distribuídos. De fato, têm-se aqui várias bibliotecas independentes e interligadas. Neste tipo de biblioteca é necessária a utilização de padrões de interoperabilidade (protocolo), garantindo a comunicação entre as diversas bibliotecas digitais e participantes do sistema.

O protocolo mais utilizado nesse tipo de arquitetura é o Z39.50 [Z39.50, 2002].

Embora existam dois tipos de arquiteturas possíveis para a construção de bibliotecas digitais, para que uma biblioteca seja considerada plenamente digital, ela deve, segundo a *Association of Research Libraries* [ARL,1995], seguir alguns critérios:

- a) A biblioteca digital não deve ser uma entidade individual.
- b) A biblioteca digital requer meios tecnológicos para conectar os recursos.
- c) As conexões entre um grande número de bibliotecas digitais e os serviços de informação devem ser transparentes para os usuários. O acesso universal às bibliotecas digitais e aos serviços de informação deve ser o objetivo principal.
- d) As bibliotecas digitais não devem limitar-se em fornecer acesso a documentos, mas devem oferecer outros elementos digitais que não podem ser fornecidos no formato impresso.

Os critérios elencados acima apontam para a arquitetura descentralizada como arquitetura ideal para a construção de uma biblioteca digital, já que a mesma não deve ser uma entidade única, não deve se limitar a armazenar documentos em apenas um formato (texto), deve prover mecanismos para que a busca da informação ocorra também em outros repositórios de informações (bibliotecas digitais).

A Fig. 1 ilustra a arquitetura descentralizada de uma biblioteca digital, exibindo várias bibliotecas digitais e permitindo o acesso e busca da informação de maneira distribuída e de forma transparente para o usuário.

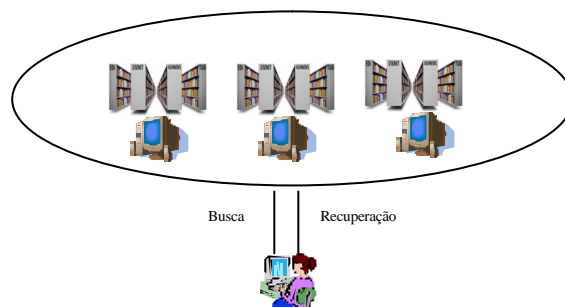


Figura 1. Arquitetura Distribuída de Biblioteca Digital

2.7 Interoperabilidade em Bibliotecas Digitais

Ao se adotar uma arquitetura centralizada na construção de uma biblioteca digital, limita-se a busca da informação ao próprio acervo da biblioteca digital. Isso não caracteriza uma situação vantajosa, pois se a informação desejada não se encontra no repositório dessa biblioteca, ela deverá ser buscada em outra biblioteca, fazendo com que a busca se torne extremamente lenta e cansativa, já que terá que ocorrer site a site.

A construção de uma biblioteca digital observando a arquitetura distribuída evita que isso aconteça, porque permite que o usuário através de um único passo (interface), possa buscar as informações desejadas nas mais diversas bibliotecas digitais sem tomar conhecimento de onde e nem como os dados estão armazenados.

Para que isso aconteça, é de suma importância que as bibliotecas digitais criadas nos mais diversos locais sejam interoperáveis. A interoperabilidade se torna um problema porque as bibliotecas digitais de uma maneira geral são heterogêneas, ou seja, elas diferem em relação à forma como armazenam e gerenciam seus dados.

A heterogeneidade pode estar ligada ao formato dos dados disponibilizados em uma biblioteca digital (.doc,.txt,.gif,.jpg), à forma de armazenamento desses dados (banco de dados, documentos XML) e ao modelo de metadados empregado na descrição dos dados (Dublin Core, Marc).

Para solucionar esse problema é necessário, portanto, a adoção de mecanismos que promovam a interoperabilidade de bibliotecas digitais.

Segundo [MARCONDES, 2001], atualmente se verificam duas maneiras de prover essa interoperabilidade. Em ambas, a idéia principal é de que o usuário interaja com uma única interface Web, de onde a busca é submetida.

A primeira é conhecida como “busca distribuída” e se caracteriza como o próprio nome, em buscar a informação (de forma transparente para o usuário) em diferentes servidores de bibliotecas digitais identificados previamente na interface do cliente como capazes de fornecer as respostas.

O segundo caminho para o tratamento da interoperabilidade, diz respeito à criação de um repositório de metadados centralizado, no qual estão armazenados os metadados dos documentos digitais de todas as bibliotecas envolvidas. É nessa base comum de metadados que é realizada a pesquisa e busca da informação de qualquer biblioteca digital.

A grande maioria das iniciativas de construção de bibliotecas digitais promovem a interoperabilidade através da primeira alternativa comentada, o que exige a presença de um protocolo apropriado para implementar a busca e recuperação da informação distribuída.

A tarefa do protocolo é, portanto, fornecer um meio para que se efetive a busca e recuperação de informação distribuída, utilizando para isso módulos apropriados, tanto no cliente (origem da busca) como no servidor (destino da busca).

Exemplos de protocolos que realizam essa tarefa são o DIENST [DIENST, 2000] e o Z39.50[Z39.50, 2002], sendo este último mais amplamente utilizado e objeto de estudo dessa dissertação.

2.8 Iniciativas Nacionais e Internacionais de Construção de Bibliotecas Digitais

Muitas iniciativas nacionais isoladas no sentido da construção de bibliotecas digitais estão sendo realizadas, principalmente por Instituições de Ensino Superior. A maioria dessas iniciativas por enquanto disponibiliza apenas periódicos e produção científica na forma de dissertações, teses e artigos.

Entre essas iniciativas pode-se destacar:

- ? Biblioteca Digital Multimídia da UFSC [BDM-UFSC, 2000]: Biblioteca digital desenvolvida pela Universidade Federal de Santa Catarina com ênfase à princípio na disponibilização do acervo das áreas de Ciências Humanas, Literatura Catarinense e obras raras da Biblioteca Universitária. Atualmente se encontra na fase de digitalização do acervo.
- ? Biblioteca Digital da Unicamp [BD-UNICAMP, 2002]: Biblioteca digital produzida pela Universidade de Campinas – UNICAMP. Disponibiliza produção científica, acadêmica e cultural e o faz oferecendo artigos, fotografias, ilustrações, teses, entre outros documentos.
- ? Biblioteca Digital de Teses e Dissertações da USP [BDTD-USP, 2001]: Biblioteca digital que disponibiliza a produção científica da comunidade universitária.

Ainda em nível nacional, cabe destacar o projeto da Biblioteca Digital Brasileira, coordenado pelo IBICT – Instituto Brasileiro de Informação, Ciência e Tecnologia, que objetiva integrar em um único portal, que pode ser acessado através do endereço www.ibict.br/bdb/portal, os mais importantes repositórios de informação digital do país, de forma a permitir consultas simultâneas e unificadas aos conteúdos informacionais destes acervos.

Como parte desse projeto, encontra-se em fase de desenvolvimento a Biblioteca Digital de Teses e Dissertações que objetiva reunir em um único repositório, o conjunto de informações relacionadas a teses e dissertações de 17 instituições acadêmicas que fazem parte desse consórcio.

A nível internacional, muitas bibliotecas digitais já estão sendo organizadas em todo o mundo com milhares de obras já digitalizadas. Entre elas pode-se citar:

- National Digital Library [NDL, 2002]: Biblioteca digital que faz parte do “National Digital Library Program” da Biblioteca do Congresso Americano disponibilizando documentos digitais sobre a história e cultura americana.

- Alexandria Digital Library [ADL, 2001]: Biblioteca digital construída pela University of Califórnia, Santa Bárbara – UCSA.

- Networked Digital Library of Theses and Dissertations [NDLTD, 2001]: reúne um consórcio internacional de universidades, bibliotecas e outras instituições com o objetivo de disponibilizar teses e dissertações eletrônicas(ETDs).

- Digital Library of MIT Theses [DLMT, 2002]: Biblioteca Digital que disponibiliza uma coleção de dissertações de mestrado e teses de doutorado de todos os departamentos do MIT – Massachuts Institute Technologic.

- Networked Computer Science Technical Reference Library [NCSTRL, 2002]: reúne um conjunto de instituições que disponibilizam relatório técnicos na área de Ciência da Computação.

2.9 Conclusão

Neste capítulo foram abordados diversos conceitos relacionados a bibliotecas digitais. Entre esses conceitos destaca-se a questão da interoperabilidade entre bibliotecas digitais, a qual pode ser obtida através de algumas abordagens, dentre elas, o uso do protocolo Z39.50.

Esse protocolo, como já foi mencionado no capítulo, é muito utilizado no contexto de bibliotecas digitais para prover a interoperabilidade de bibliotecas digitais e será o assunto do próximo capítulo.

Capítulo 3 Protocolo Z39.50

O ANSI/NISO Z39.50 faz parte de um conjunto de padrões produzidos para facilitar a interconexão entre sistemas de computador.

O Z39.50 é um protocolo do nível de aplicação que permite a comunicação cliente/servidor entre dois computadores com o objetivo de buscar e recuperar informações. Ele especifica procedimentos e estruturas para um cliente pesquisar um banco de dados mantido por um servidor, recuperar registros do banco de dados identificados por uma pesquisa e ordenar um conjunto de resultados.

Atualmente ele é considerado o protocolo padrão para busca de informação distribuída no contexto das bibliotecas digitais.

O objetivo desse capítulo é apresentar uma revisão de vários tópicos relacionados a este protocolo, para que isso forneça a fundamentação necessária para o entendimento da proposta dessa dissertação.

Os tópicos apresentados vão desde o histórico de desenvolvimento, modelo de funcionamento, definição dos serviços oferecidos até o formato das mensagens trocadas entre cliente e servidor.

3.1 Histórico de Desenvolvimento

O padrão Z39.50 foi proposto originalmente em 1984 pela NISO - National Information Standards Organization, uma organização participante da ANSI - American National Standards Institute, para prover um modo padrão de pesquisar bancos de dados bibliográficos. Em 1988 foi aprovada a primeira versão, em 1992 a segunda versão e em 1995 a terceira, sendo esta a versão que ainda é utilizada.

Atualmente o Z39.50 é mantido pela Z39.50 Maintenance Agency [Z39.50 M.A., 2001] administrada pela *Library of Congress* [LOC, 2002] dos Estados Unidos, que é a agência oficial responsável pela manutenção, atualização, coordenação técnica e pelo desenvolvimento deste padrão de recuperação de informação.

3.2 Modelo de Funcionamento do Z39.50

Z39.50 define um modo padrão para dois computadores se comunicarem com a finalidade de recuperação de informação. Através da padronização de procedimentos e características, Z39.50 torna fácil a tarefa de pesquisar e recuperar informações em grandes bancos de dados. Especificamente, Z39.50 suporta recuperação de informação dentro um ambiente cliente/servidor distribuído, onde um computador que opera como um cliente submete um pedido de busca (uma consulta) para outro computador que age como um servidor de informação. Um software no servidor executa a busca em um ou mais bancos de dados e cria um conjunto de resultados composto de registros que atendem aos critérios do pedido de busca. O servidor retorna para o cliente o conjunto de resultados contendo registros para que o mesmo possa processá-los. A Fig. 2 ilustra essa comunicação.

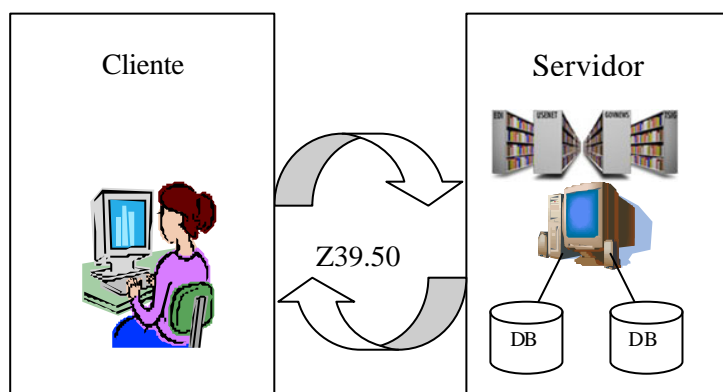


Figura 2. Comunicação entre Cliente e Servidor através do Protocolo Z39.50

O poder de Z39.50 é que ele separa a interface de usuário no lado cliente dos servidores de informação, mecanismos de busca e bancos de dados. Z39.50 provê uma visão consistente da informação a partir de uma ampla variedade de fontes, e oferece para os implementadores de cliente a capacidade de integrar informações a partir de um grande número de bancos de dados.

A seguir é apresentada com detalhes a comunicação entre um cliente e um servidor Z39.50:

A comunicação entre cliente e servidor acontece através de uma *Z-Association*. Uma *Z-Association* é explicitamente estabelecida pelo cliente e pode ser terminada explicitamente pelo cliente ou servidor, ou implicitamente terminada através da perda de conexão.

Os papéis de cliente (*Z-Client*) e servidor (*Z-Server*) dentro de uma *Z-Association* não podem ser invertidos. Uma *Z-Association* não pode ser reiniciada, assim uma vez que uma *Z-Association* é terminada nenhuma informação de estado é retida, exceto a informação que é explicitamente salva.

O itinerário de um processo de busca envolvendo este protocolo se dá da seguinte forma:

- ? Usuário seleciona a biblioteca destino (*Z-Server*) a partir de um menu de uma aplicação;
- ? Usuário entra com os termos de pesquisa.
- ? Os termos de pesquisa e a biblioteca destino são enviados para o *Z-Client*
- ? *Z-Client* traduz os termos pesquisados para dentro de “*Z-Speak*” e contata o software *Z-Server* da biblioteca destino.
- ? Há uma negociação preliminar entre *Z-Client* e *Z-Server* para estabelecer as regras para a “*Z-Association*” entre os dois sistemas.
- ? *Z-Server* traduz o “*Z-speak*” para dentro de uma requisição de busca para o banco de dados da biblioteca destino e recebe a resposta sobre o número de comparações encontradas.
- ? *Z-Client* recebe os registros.
- ? Registros são apresentados para o usuário.

A Fig. 3 demonstra esse itinerário:

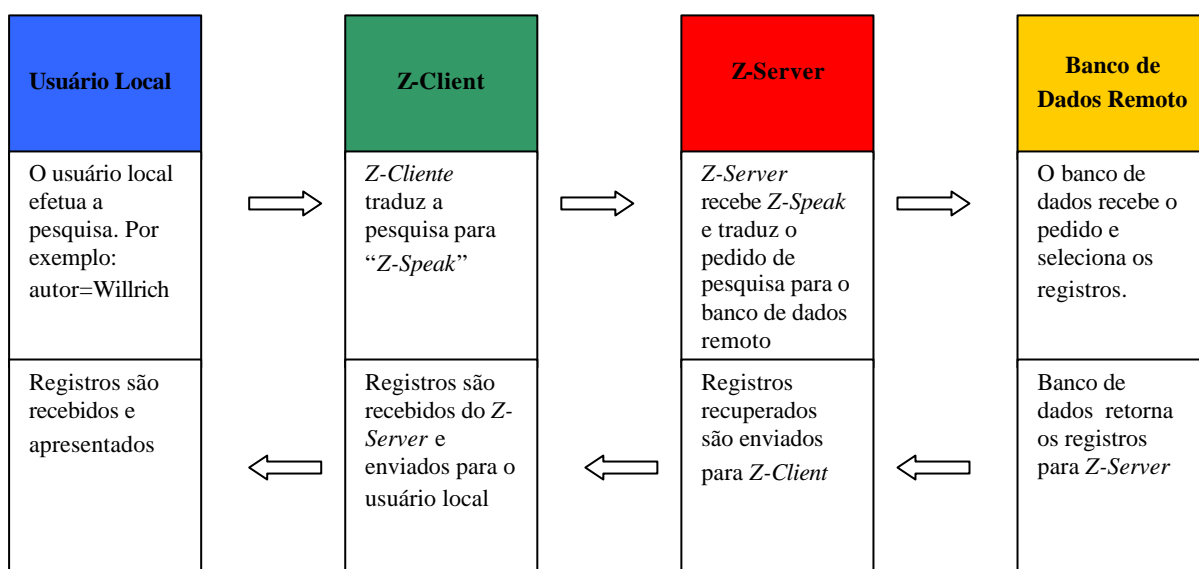


Figura 3. Itinerário de Busca de Informação através da Utilização do Protocolo Z39.50

3.3 Provendo a Interoperabilidade

Z39.50 reconhece que a recuperação de informação consiste em dois componentes primários: seleção de informação baseada em algum critério e recuperação dessa informação, e provê uma linguagem comum para ambas as atividades. Z39.50 padroniza a maneira pela qual um cliente e um servidor se comunicam e promove a interoperabilidade quando há diferenças entre os sistemas de computador, mecanismos de procura e bancos de dados.

A interoperabilidade é alcançada através da padronização de:

- ? -Mecanismos de Codificação - um modo padrão de codificar os dados para remessa ao longo da rede.
- ? -Conteúdo Semântico - um modelo de dados padrão com conhecimento semântico compartilhado por comunidades específicas permite a busca e recuperação interoperável dentro de cada um desses domínios.

3.3.1 Mecanismos de Codificação

Z39.50 define um protocolo da camada de aplicação do modelo OSI e requer um serviço de transporte seguro como TCP (Z39.50 tem um número de porta TCP registrado). Ele especifica formatos e procedimentos que controlam a troca de mensagens entre um cliente ou "origem" e servidor ou "destino". As mensagens, que tecnicamente são conhecidas como "unidades de dados de protocolo de aplicação" ou APDU, são enviadas entre um cliente e um servidor e são especificadas em Notação de Sintaxe Abstrata 1 (ASN.1), uma sintaxe que descreve a estrutura da APDU. As Regras de Codificação Básicas ou *Basic Encoding Rules* (BER) são usadas para serializar as estruturas ASN.1.

IMPLEMENTAÇÃO BÁSICA DO Z39.50

A maioria das implementações do Z39.50 são realizadas com a utilização de um dos toolkits de Z39.50 disponíveis, e começam de maneira simplificada com a verificação da interoperabilidade em relação a outras aplicações e posteriormente incrementam funcionalidades com características Z39.50 adicionais.

Uma implementação básica de Z39.50 é composta de serviços que negociam a inicialização entre o cliente e servidor, executam uma busca em um banco de dados, criam um conjunto de resultados composto de registros de banco de dados que atendem ao critério da busca e recuperam um ou mais registros do conjunto de resultados.

A seguir é demonstrado o funcionamento de uma implementação básica da perspectiva do usuário e do Z39.50.

A Perspectiva do Usuário

Em uma implementação básica, como mostrado na Fig. 4, o usuário escolhe uma biblioteca, entra com os critérios de busca, submete a busca e são retornados e exibidos os registros que correspondem a busca. O usuário pode então optar por acessar o documento a partir da localização do mesmo que consta no registro retornado.

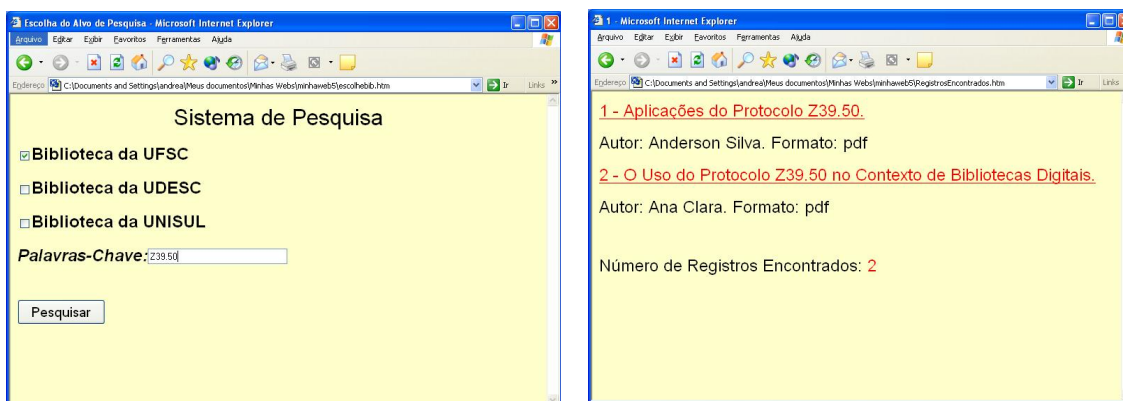


Figura 4. Implementação do Z39.50 da Perspectiva do Usuário

A Perspectiva do Z39.50

Da perspectiva do Z39.50, como mostra a Fig. 5, uma série de mensagens trocadas entre o cliente e servidor estabelecem uma conexão, inicializam uma sessão Z39.50 e negociam expectativas e limitações nas atividades que acontecerão por exemplo: tamanho de máximo dos registros que serão transferidos do servidor ao cliente, a versão do protocolo suportada, etc. (definida pela *Initialization Facility* através do serviço *Init*, que será explicado posteriormente). Depois que estes acordos são negociados, o cliente pode submeter uma busca. O cliente de Z39.50 traduz a busca em uma representação padronizada e passa ela para um servidor Z39.50 (definida pela *Search Facility* através do serviço *Search*). O servidor executa a busca em um banco de dados, e um conjunto de registros, chamado de "conjunto de resultados" é mantido no servidor. O resultado da busca é um relatório do número de registros que estão inclusos no conjunto de resultados. O cliente pode pedir por registros a partir do conjunto de resultados (definida pela *Retrieval Facility* através dos serviços *Present* e *Segment*) ou pedir processamento adicional ao conjunto de resultados. Após a recepção dos registros, o cliente pode processar e exibir os registros ao usuário.

O protocolo define várias linguagens de pesquisa para especificar a busca ou pesquisa e várias sintaxes de registro que podem ser usadas para transferir registros do servidor para o cliente.

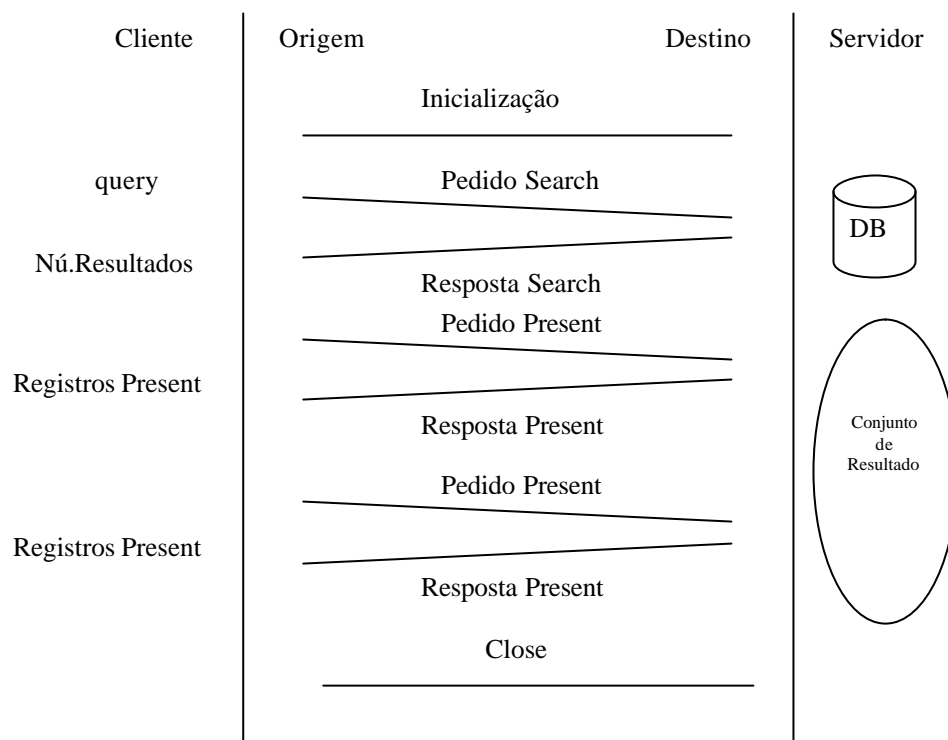


Figura 5. Implementação da Perspectiva do Z39.50

FUNCIONALIDADES EXTRAS

Uma implementação de Z39.50 pode ser adicionada de funcionalidades como a definição de mensagens para recurso e gerenciamento de acesso. Por exemplo, um servidor pode prover relatórios de progresso para uma busca ativa, ou pode pedir para o cliente autorização para continuar a busca intensiva por um recurso; um cliente pode abortar uma busca ativa.

Z39.50 também define mensagens para gerenciar um conjunto de resultados, para ordenar um conjunto de resultados, para abrir e fechar conexões, e também um mecanismo chamado "*Extended Services*", que é essencialmente um mecanismo de chamada de procedimento remoto assíncrono que o cliente pode usar para invocar serviços no servidor. "*Extended Services*" provêem meios de salvar conjunto de resultados por sessões, enfileirando-os para impressão ou processamento de correio eletrônico no servidor, ou para registrar e gerenciar consultas que seriam executadas periodicamente no servidor.

3.3.2 Conteúdo Semântico

Junto com a padronização da codificação de mensagens está o conceito de "conhecimento semântico compartilhado". Várias classes ou domínios de conteúdo de informação foram estabelecidos por consenso da comunidade para prover um entendimento compartilhado da estrutura e atributos dentro daquele domínio. Isto habilita acesso uniforme a fontes de informação heterogêneas dentro de um domínio.

O modelo arquitetônico básico de Z39.50 gira ao redor deste conceito de conteúdo semântico. O servidor apresenta uma visão abstrata baseada em registros de seu banco de dados dentro de um domínio semântico, isto é, uma representação de um banco de dados virtual que contém registros com a estrutura lógica do banco de dados "real" que é escondido.

Associado com cada banco de dados virtual está um conjunto de pontos de acesso (ou atributos de pesquisa) que podem ser usados para procurar e um conjunto de pontos de recuperação (ou elementos de Schema) para apresentar os dados de volta ao usuário. Dentro de um determinado domínio, pontos de acesso são registrados em "*Attribute Sets*" ou Conjunto de Atributos, enquanto que pontos de recuperação estão definidos em *Schemas* padronizados.

Z39.50 tem sido adotado amplamente para prover acesso a muitas classes de informação com "*Attribute Sets*" e *Schemas* registrados. Isto inclui, mas não está limitado a: dados bibliográficos, recursos de informação de governo (nacionalmente e internacionalmente), dados científicos e técnicos, dados geo-espaciais.

3.4 Definição de Serviços do Protocolo

O protocolo Z39.50 apresenta uma série de serviços que são divididos em 11 grupos lógicos ou categorias, conhecidas como "*facilities*". Cada grupo é composto de um ou mais serviços.

Abaixo é apresentada uma breve descrição de todos os serviços dentro de seus grupos. A descrição detalhada se encontra no Anexo 1.

- ? **Initialization Facility:** É composto pelo serviço *Init*, que permite ao cliente negociar a inicialização com o servidor (Z-Associação). Isso inclui as características suportadas por Z39.50 como: conjunto de caracteres default, linguagem default, versão de protocolo e autenticação de usuário.
- ? **Search Facility:** É composto da primitiva *Search*, que permite ao cliente pesquisar bancos de dados no servidor usando um formato de procura bem conhecido, criar um conjunto de resultados composto de registros no servidor e receber informação sobre este conjunto de resultados. A pesquisa pode conter operadores booleanos, termos de pesquisa, etc.
- ? **Retrieval Facility:** composto pelos serviços: *Present* e *Segment*. O serviço *Present* permite ao cliente pedir um ou mais registros a partir de um conjunto de resultados especificado. Isso inclui pedir faixas específicas de resultados (por exemplo, 10 de 20 registros), elementos específicos em um registro (por exemplo, título e autor). O serviço *Segment* é um serviço iniciado pelo servidor, durante um serviço *Present*. Se os registros pedidos por um pedido *Present* não se ajustarem em um único segmento, e se a segmentação estiver em uso, o servidor devolve múltiplos segmentos, cada um contendo uma porção dos registros. Um serviço *Present* consiste de um pedido *Present* seguido de zero ou mais pedidos *Segment* seguido de uma resposta *Present*.
- ? **Result-set-delete Facility:** É composto pelo serviço *Delete*, que permite um cliente pedir a deleção de conjuntos de resultados específicos ou de todos os conjuntos de resultados.
- ? **Browse Facility:** É composto pelo serviço *Scan*, que é usado para examinar uma lista de termos ordenada (assunto, nomes, títulos, etc.) pelo servidor.
- ? **Sort Facility:** É composto pelo serviço *Sort*, que permite a um cliente pedir para o servidor ordenar um conjunto de resultados (ou unir múltiplos conjuntos de resultado e então ordená-los).
- ? **Access Control Facility:** É composto pelo serviço *Access-control*, que permite um servidor recusar um cliente. A recusa poderia pertencer a uma operação

específica ou a uma Z-Associação. O mecanismo de pedido/resposta de Access-control pode ser usado para suportar controle de acesso de recusa ou autenticação, incluindo recusas de senha, chave pública de sistemas criptografados e autenticação algorítmica. Ele não inicia uma operação, e ele pode ou não ser parte de uma operação ativa.

? **Accounting/Resource Control Facility** composto por três serviços: *Resource-control*, *Trigger-resource-control* e *Resource-report*.

O serviço *Resource-control* iniciado pelo servidor. Ele permite ao servidor enviar um pedido de *Resource-control*, ou seja, notificar o cliente que o consumo de recursos previstos ou atual excederam os limites permitidos.

O serviço *Trigger-resource-control* permite ao cliente pedir que o servidor inicie um serviço de *Resource-control* ou cancele a operação atual.

O serviço *Resource-report* permite ao cliente pedir um relatório de recursos pertencente a uma operação completada ou a uma Z-Associação.

? **Explain Facility** *Explain facility* não inclui qualquer serviço, mas usa os serviços da *Search and Retrieval facilities*. É usada para descobrir detalhes de implementação do servidor, incluindo características gerais (descrição, informação de comunicação, horas de operação, restrições, custo de uso, etc.), bancos de dados disponíveis para pesquisa, conjunto de atributos, detalhes de atributo, schemas, sintaxes de registro, especificações de elemento e serviços estendidos suportados.

? **Extended Services Facility:** É composto por um único serviço: *Extended-services*, o qual permite acesso a serviços fora do protocolo, que podem continuar depois que a Z-Associação for terminada. Os serviços estendidos definidos por este padrão incluem agendamento de pesquisas periódicas e atualização de banco de dados.

? **Termination Facility:** composto pelo serviço *Close*, o qual pode ser iniciado pelo cliente ou servidor. Ele permite que um cliente ou servidor termine todas as

operações ativas e inicie o término da Z-Associação. (seguindo o termino da Z-Association, o cliente pode tentar inicializar outra Z-Associação que usa o serviço *Init*)

3.5 Especificação da Transferência de Informação no Protocolo

O protocolo de aplicação de recuperação de informação Z39.50 especifica os formatos e procedimentos que governam a transferência de informação entre um cliente e servidor Z39.50.

A comunicação entre um cliente e servidor é feita através da ADPU (*Unit Data Protocol Application*), que é uma unidade de informação cujo formato é estabelecido pelo protocolo Z39.50 e consiste de informação de protocolo de aplicação e de dados de aplicação de usuário.

3.6 Conclusão

Neste capítulo foi apresentado uma descrição de vários tópicos pertinentes ao protocolo Z39.50, com o objetivo de se obter um maior conhecimento a cerca deste protocolo. Entre os tópicos apresentados constam o histórico de desenvolvimento do Z39.50, o modelo de funcionamento, como ele promove a interoperabilidade e a definição dos seus principais serviços.

Esse conhecimento é necessário para o entendimento da proposta de extensão do protocolo Z39.50, através da criação de um novo serviço, que será apresentado no próximo capítulo.

Capítulo 4 Extensão do Z39.50

O protocolo Z39.50 contempla a comunicação entre um cliente (implementa o módulo cliente do protocolo) e um servidor (implementa o módulo servidor do protocolo), sendo que o endereço de um ou mais servidores (destino da busca) devem ser previamente conhecidos e disponibilizados na aplicação do cliente em tempo de codificação.

Entende-se que a necessidade do conhecimento prévio e posterior disponibilização via interação humana através da codificação da aplicação se caracteriza numa limitação, pois torna o processo de escolha de bibliotecas digitais a serem pesquisadas estático.

A proposta dessa dissertação é evitar que essa interação humana aconteça, possibilitando que uma lista com informações como nome e endereço de servidores de bibliotecas digitais que suportam o protocolo Z39.50 seja construída dinamicamente dentro da aplicação cliente. Para que isso se torne possível, está se propondo a criação de um novo serviço que irá se unir aos serviços já existentes do protocolo Z39.50 e permitirá a busca de informações (nome e endereço) de servidores de bibliotecas digitais. Ele se encarregará de buscar as informações de servidores de bibliotecas digitais e irá disponibilizar as mesmas para a aplicação do cliente, caracterizando com isso a construção dinâmica da lista com informações de servidores.

Esse novo serviço chamado *Searchlib* deve ser o primeiro a ser disparado pelo cliente no caso do usuário querer efetuar uma busca por servidores de bibliotecas digitais. No caso do usuário querer efetuar uma pesquisa por documentos digitais diretamente na base de dados do servidor que a aplicação do cliente conhece o endereço (porque é requisito que ela conheça pelo menos um endereço de servidor de biblioteca digital), o serviço de busca de bibliotecas não precisa ser disparado e o processo de pesquisa é desenvolvido normalmente pelos outros serviços que fazem parte do protocolo Z39.50.

Várias alternativas de busca de endereços de servidores de bibliotecas digitais envolvendo a utilização do novo serviço proposto, foram elaboradas e analisadas, e uma

síntese de cada uma delas será apresentada na seção que segue. Em seguida, a solução escolhida será detalhada.

4.1 Descrição e Avaliação das Alternativas

Como mencionado acima, o serviço *Searchlib* possui o objetivo de buscar endereços de servidores de bibliotecas digitais que implementam o Z39.50. Para tal, existem diversas alternativas de implementar este serviço. As alternativas analisadas foram:

- ? Busca Centralizada
- ? Busca Distribuída a partir do Servidor
- ? Busca a partir do Cliente

4.1.1 Busca Centralizada

Nessa proposta alguns servidores de bibliotecas digitais são escolhidos para manterem uma tabela contendo informações como nome e endereço de todas as bibliotecas digitais que suportam o protocolo Z39.50. Esses servidores são conhecidos como “servidor de nomes”, justamente por manterem uma tabela contendo informações sobre outros servidores.

De acordo com a premissa básica da proposta, o cliente conhece previamente o endereço de apenas um servidor de biblioteca digital e o pedido de serviço de busca de bibliotecas digitais ou *Searchlib* é direcionado para esse servidor. O servidor recebe o pedido e retorna na resposta do serviço o endereço de um ou mais “servidores de nomes”. O cliente recebe a resposta e reencaminha o pedido de busca de bibliotecas automaticamente para o endereço desse “servidor de nomes”. O “servidor de nomes” recebe o pedido e retorna uma resposta com o nome e endereço de todos os servidores de bibliotecas digitais que constam em sua tabela.

A existência de mais de um “servidor de nomes” visa criar níveis de redundância: caso um servidor não esteja acessível, o cliente poderá encaminhar seu pedido para outro servidor.

O endereço destes “servidores de nomes” devem ser conhecidos pelos clientes Z39.50 ou informados para eles por qualquer servidor. Portanto, qualquer servidor de biblioteca digital deve manter a informação do endereço de um ou mais “servidores de nomes”.

Esta alternativa obviamente possui grandes desvantagens que são:

- ? **Dificuldade de administração da tabela de nomes:** toda criação de uma nova biblioteca Z39.50 ou a alteração tanto do nome como do endereço de um servidor de biblioteca digital requer uma atualização das tabelas de todos os “servidores de nomes”. Como isto deve ser feito por uma entidade centralizadora pode demorar um certo tempo.
- ? **Grande número de registros retornados pelo servidor:** o número de registros contendo nome e endereço de servidores de bibliotecas digitais, mantidos na tabela do “servidor de nomes” pesquisado deverá ser muito grande, o que ocasionará também um grande número de registros retornados na resposta para o cliente. A aplicação do cliente deve receber as respostas e exibir a lista com nome e endereços de todos os servidores para que o usuário então possa escolher em qual ou em quais servidores de bibliotecas ele fará a pesquisa. Isso torna o processo tanto em nível de exibição quanto de escolha do usuário um tanto demorado, já que serão muitos nomes de servidores de bibliotecas digitais que aparecerão na tela do usuário.

4.1.2 Busca Distribuída a partir do Servidor

Nesta segunda proposta cada servidor de biblioteca digital possui uma tabela contendo nome e endereço de um ou mais servidores de bibliotecas digitais, porém não de todos os servidores, existindo a proposta de distribuição do cadastramento de nomes e endereços de servidores em todos os servidores de bibliotecas digitais.

Nesta solução, um cliente emite o pedido de busca de bibliotecas digitais a um servidor cujo endereço é conhecido na aplicação cliente e este servidor retorna para o cliente o nome e endereço de servidores da sua tabela e também reencaminha novos pedidos de busca para todos os servidores da sua tabela. Assim, o servidor é o

responsável pelo encaminhamento de novos pedidos de busca de biblioteca para todos os servidores que existem na sua tabela.

A busca distribuída a partir do servidor pode ser implementada de duas formas: Encadeada ou Hierárquica.

ENCADEADA

A característica principal desta proposta reside na maneira como um servidor de biblioteca digital é cadastrado na tabela de outro servidor de biblioteca digital, existindo uma espécie de “encadeamento”, na forma de uma lista encadeada ou anel.

A Fig. 6 ilustra essa organização encadeada, onde cada servidor armazena o nome e endereço de apenas um servidor de biblioteca digital.

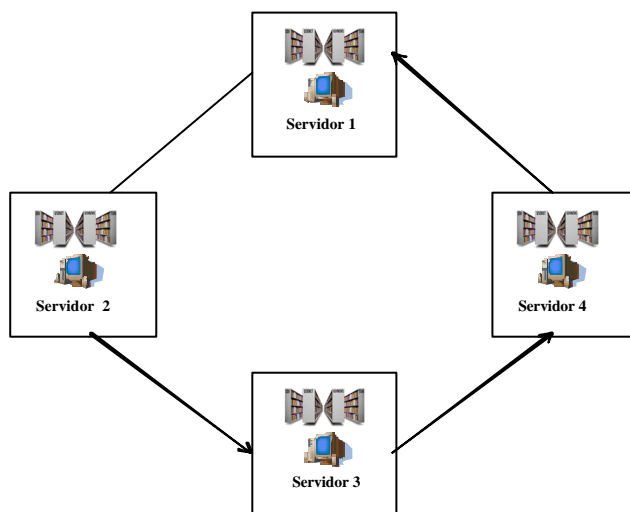


Figura 6. Organização Encadeada de Servidores de Bibliotecas Digitais

A Fig. 7 ilustra a inclusão de um novo servidor na estrutura. Nesta figura, o servidor 5 desejando participar da estrutura, solicita ao servidor 3 a sua inclusão. O servidor 3 modifica o cadastro da sua tabela (que contem as informações sobre o servidor 4) para “apontar” para novo servidor da estrutura e informa para este novo servidor as informações sobre o servidor 4 que ele “apontava”, para que o mesmo possa cadastrar as informações em sua tabela. No

final da inclusão do novo servidor na estrutura, o servidor 3 “aponta” para o servidor 5 que “aponta” para o servidor 4.

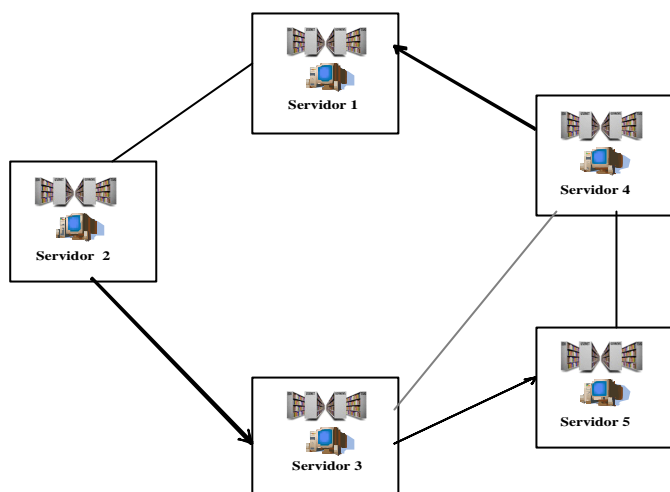


Figura 7. Inclusão de um Novo Servidor na Organização Encadeada

Nessa proposta, o procedimento de busca de bibliotecas digitais opera da seguinte forma:

- ? O cliente emite o pedido de busca de bibliotecas ou *Searchlib* para o servidor que está configurado na aplicação;
- ? O servidor recebe o pedido e retorna a resposta com nome e endereço do servidor de biblioteca digital que está cadastrado em sua tabela e também reencaminha o pedido para este mesmo servidor. Este passo se repete até que o pedido chega novamente ao servidor ao qual o cliente se conectou. Esse servidor verifica o endereço do cliente que originou o pedido e pára o processo de busca.

À medida que as respostas vão chegando para o cliente, o mesmo repassa para a aplicação e esta exibe para o usuário. Após a exibição do nome de todos os possíveis servidores de bibliotecas digitais encontrados, o usuário pode escolher um ou mais

servidores de bibliotecas digitais e prosseguir com a pesquisa por documentos digitais através da digitação de palavras-chave.

Essa proposta é similar ao funcionamento do protocolo *peer-to-peer* Gnutella, pois torna o servidor também um cliente, no sentido que além de oferecer recursos, no caso da base de dados para pesquisa, ele também encaminha pedidos de busca de bibliotecas digitais. Além disso, o mecanismo de busca dessa proposta é simular a pesquisa por recursos do protocolo Gnutella, pois um servidor ou nó recebe um pedido, retorna uma resposta e reencaminha esse pedido para outros servidores ou nós que fazem o mesmo processo.

A proposta apresentada também possui claras desvantagens:

- ? **Pouca tolerância à falhas:** como existe um encadeamento entre os servidores de bibliotecas digitais, caso ocorra a situação de um servidor falhar, a busca por servidores biblioteca digital falha, já que um servidor sempre “aponta” para outro servidor e assim por diante.
- ? **Incremento das atividades de processamento do servidor:** como esse tipo de arquitetura propõe que o servidor reencaminhe os pedidos de busca de biblioteca, isso ocasiona um incremento das atividades de processamento do servidor, fazendo praticamente que ele se torne ao mesmo tempo um cliente, já que ele também vai emitir pedidos de busca de biblioteca.
- ? **Grande número de registros retornados pelos servidores:** assim como na arquitetura centralizada, um grande número de respostas contendo nome e endereço de servidores de bibliotecas digitais é retornado para o cliente. Isso se torna o processo tanto a nível de exibição quanto de escolha do usuário um tanto demorado, já que serão muitos nomes de servidores de bibliotecas digitais que aparecerão na tela do usuário. Portanto, a busca por bibliotecas se torna mundial, não havendo como delimitar um determinado escopo para que a busca de bibliotecas ocorra.

HIERÁRQUICA

Nesta solução é proposta a criação de uma organização hierárquica dos servidores de bibliotecas digital. Os servidores são organizados logicamente em escopos que são: local, regional, nacional ou global.

O escopo local existe no contexto de uma instituição, portanto, os servidores pertencentes a essa instituição possuem um “escopo local”, sendo chamados de **“servidores locais”**. Entre esses servidores deve ser escolhido um ou mais representantes de servidores locais que são chamados de **“servidor de nomes local”**. Na tabela de um “servidor local” é cadastrado nome e endereço do “servidor de nomes local”. Na tabela de um “servidor de nomes local” são cadastrados o nome, endereço e escopo (neste caso sempre local) de todos os servidores locais da instituição em questão e obrigatoriamente o nome, endereço e escopo (regional) de um ou mais “servidores de nomes regional”.

O escopo regional se dá no contexto de uma determinada região que pode englobar alguns municípios, um estado ou mais de um estado. Deve ser escolhido entre os servidores que são “servidor de nomes local” um representante que será chamado de **“servidor de nomes regional”**. Na tabela de um “servidor de nomes regional” é cadastrado nome, endereço e escopo (neste caso todos terão escopo regional) de todos os servidores designados “servidor de nomes local”, ou seja, os representantes das diversas instituições que o escopo regional engloba e obrigatoriamente o nome, endereço e escopo (nacional) de um ou mais “servidor de nomes nacional”. Além disso, este servidor também irá conter o nome, endereço e escopo (local) de “servidores locais”, já que o “servidor de nomes regional” é escolhido entre os “servidores de nomes locais”.

O escopo nacional acontece no contexto de um país e um **“servidor de nomes nacional”** deve ser escolhido entre os “servidores de nomes regionais”. Na tabela de um “servidor de nomes nacional” é cadastrado nome, endereço e escopo (neste caso todos terão escopo nacional) de todos os servidores designados “servidor de nomes regional”, ou seja, os representantes das diversas regiões que o escopo nacional engloba e obrigatoriamente o nome, endereço e escopo (global) de um ou mais “servidor de nomes

global” Além disso, este servidor também irá conter o nome, endereço e escopo (regional) dos “servidores de nomes locais”, já que o “servidor de nomes nacional” é escolhido entre os “servidores de nomes regional” existentes no país. Ele também conterá nome, endereço e escopo (local) de servidores locais, pois ele se ele é “servidor de nomes regional” é porque também é “servidor de nomes local”.

Finalmente, o escopo global acontece à nível mundial e o “**servidor de nomes global**” deve ser escolhido entre os “servidores de nomes nacionais”. Na tabela de um “servidor de nomes global” é cadastrado nome, endereço e escopo (global) de todos os servidores designados “servidor de nomes nacional”, ou seja, os representantes dos diversos países que o escopo global engloba. Além disso, este servidor também irá conter também o nome, endereço e escopo (nacional) dos “servidores de nomes regional”, o nome, endereço e escopo (regional) dos “servidores de nomes local” e o nome, endereço e escopo (local) dos servidores locais. Isso acontece porque um “servidor de nomes global” também é servidor de nomes nacional, regional e local.

Junto com a tabela que contém informações sobre servidores de bibliotecas digitais, cada “servidor de nome” deve manter uma variável indicando o “**tipo de servidor**” que ele é: servidor local, servidor de nomes local, servidor de nomes regional, servidor de nomes nacional ou servidor de nomes global.

Nesta proposta, o procedimento de busca de servidores de bibliotecas digitais é o seguinte:

- ? Um cliente deve submeter um pedido de busca de bibliotecas digitais ou *Searchlib* a um servidor de biblioteca digital Z39.50 indicando o escopo da busca: nenhum, local, regional, nacional ou global.
- ? Se o “tipo de servidor” do servidor for “servidor local”, ou seja, o servidor não é um servidor de nomes, o pedido de *Searchlib* é encaminhado automaticamente pelo servidor para o endereço do “servidor de nomes local”.

- ? Se o servidor for do tipo “servidor de nomes”, que pode ser local, regional, nacional ou global, ele deve analisar o escopo do pedido (escolhido pelo usuário) e a partir disso, as seguintes situações podem acontecer:
- o Se o escopo do pedido é igual ou inferior ao “tipo de servidor”: servidor retorna para o cliente que emitiu o pedido *Searchlib*, os registros contendo nome, endereço e escopo de todos os servidores que estão cadastrados em sua tabela cujo campo escopo é igual ao escopo do pedido. A situação do escopo do pedido ser inferior acontece, por exemplo, quando um pedido de busca de bibliotecas local é feito para um “servidor de nomes nacional”. Nesse caso ele pode atender, porque se ele é do tipo “servidor de nomes nacional” é escolhido entre os “servidores de nomes regionais” que por sua vez são escolhidos entre os “servidores de nomes locais”.
 - o Se o escopo do pedido for superior “tipo de servidor”: o “servidor de nomes” deve encaminhar o pedido para o servidor que está cadastrado em sua tabela cujo campo escopo for superior ao seu tipo. Esse processo deve se repetir até que o pedido *Searchlib* encontre um servidor que atenda ao seu escopo. Na satisfação da condição, o servidor retorna para o cliente que emitiu o pedido *Searchlib*, os registros contendo nome, endereço e escopo de todos os servidores cujo campo escopo é igual ao escopo do pedido.

Esse tipo de arquitetura possui como principal desvantagem o incremento das atividades de processamento do servidor, já que os novos pedidos de busca de bibliotecas são efetuados a partir do servidor. Mas ela apresenta uma vantagem em relação às arquiteturas anteriores, pois possibilita que o usuário possa utilizar critérios, ou seja, escolha um escopo no momento do pedido de busca de bibliotecas, evitando com isso que um grande número de registros de contendo informações de servidores retorne para o cliente.

4.1.3 Busca a partir do Cliente

Nesta proposta, a idéia de distribuição do cadastramento de informações sobre servidores permanece, ou seja, cada servidor de biblioteca digital possui uma tabela contendo nome e endereço de outros servidores de bibliotecas digitais. A diferença em relação às arquiteturas anteriores é que a emissão de pedidos de busca de bibliotecas digitais ou *Searchlib* agora é tarefa totalmente do cliente, ou seja, um pedido é feito para um servidor, que retorna uma resposta para o cliente contendo registros com nome e endereço de outros servidores e a partir do cliente, se for o caso, novos pedidos de busca de biblioteca são encaminhados para outros servidores.

Retira-se, portanto, a tarefa do servidor de encaminhar novos pedidos de busca.

Diante dessa abordagem, duas propostas foram estudadas e analisadas: Hierárquica e *Multicast*.

HIERÁRQUICA

Esta proposta é praticamente idêntica à proposta Hierárquica da Busca Distribuída a partir do Servidor apresentada anteriormente. O que difere é que aqui o pedido de busca de bibliotecas ou *Searchlib* é feito sempre a partir do cliente, ou seja, ao invés do servidor reencaminhar o pedido para outro servidor, quando for o caso, ele deve retornar como resposta o nome e endereço desse servidor para que o cliente possa então encaminhar ele mesmo o pedido de busca.

Essa solução não apresenta a desvantagem de incrementar as atividades de processamento do servidor, já que os novos pedidos de busca de bibliotecas são efetuados a partir do cliente e possibilita também que o usuário possa utilizar critérios na busca de bibliotecas, evitando com isso que um grande número de registros de servidores retorne para o cliente e possibilitando que o usuário também controle a busca de biblioteca até chegar a um servidor local de biblioteca digital.

MULTICAST

Na proposta denominada *Multicast* ou uso de mensagem de difusão seletiva, prevê-se a alocação de um ou mais endereços IP *multicast* para a submissão do pedido de *Searchlib* para um grupo *multicast* formado por todas as bibliotecas digitais Z39.50.

Sendo assim, o pedido do cliente alcança todas as bibliotecas digitais, sendo que elas informaram sua existência retornando nome e endereço ao cliente.

As principais desvantagens da utilização dessa solução são:

- ? Inviabilidade de implantação na estrutura de rede Internet atual: Como o *multicast* não é suportado pela maioria dos roteadores na Internet, esta proposta não poderia ser implantada na Internet atual. Com uma futura adoção do protocolo Ipv6, esta solução poderia ser uma alternativa viável;
- ? Incremento do tráfego na rede: como a primitiva *Searchlib* deveria alcançar todos os servidores Z39.50 do mundo, e todos estes deveriam responder este serviço, o tráfego na rede teria um incremento relevante;
- ? Impossibilidade de organização complexa: caso se queira realizar uma organização hierárquica das bibliotecas, deve-se atribuir um endereço IP *multicast* para cada nível. Com isto, terá que ser alocado uma faixa grande de endereços IP *multicast* para este tipo de aplicação, e os clientes deveriam conhecer o endereço *multicast* de cada país, região, instituição, etc, ou então a existência de um serviço que fornecesse esta informação.

4.1.4 Análise das Alternativas

Na apresentação de cada alternativa envolvendo a utilização do serviço *Searchlib* foram destacadas as principais desvantagens de cada uma e algumas vantagens. A seguir é apresentado na Tabela 1 um resumo das desvantagens e posteriormente é feita a análise da melhor alternativa.

Alternativas	Desvantagens
Centralizada	<ul style="list-style-type: none"> ? Dificuldade de administração da tabela de nomes ? Grande número de registros retornados pelo servidor
Distribuída a partir do Servidor, com Organização Encadeada	<ul style="list-style-type: none"> ? Pouca tolerância a falhas Incremento das atividades de processamento do servidor. ? Grande número de registros retornados pelos servidores

Distribuída a partir do Servidor, com Organização Hierárquica	? Incremento das atividades de processamento do servidor
A partir do Cliente em Multicast	? Inviabilidade de implantação na estrutura de rede Internet atual ? Incremento do tráfego na rede ? Impossibilidade de organização complexa
A partir do Cliente, com Organização Hierárquica	

Tabela 1. Desvantagens das Alternativas de Busca de Bibliotecas Digitais

Analisando as alternativas apresentadas, conclui-se que a Busca a partir do Cliente, com Organização Hierárquica é a que se constitui na melhor proposta, otimizando o processo de busca de servidores de bibliotecas digitais. Essa alternativa possui as seguintes vantagens em relação às alternativas anteriormente apresentadas:

- ? A informação contendo nome e endereço de servidores de bibliotecas digitais não se encontra cadastrada de forma centralizada em poucos servidores, ficando distribuída em vários servidores de bibliotecas digitais. Isso propicia que a manutenibilidade das tabelas se torne mais facilitada, principalmente no sentido da criação de novos servidores de bibliotecas digitais. Por exemplo: Em uma instituição que já possui servidores de bibliotecas digitais, mais um servidor é criado. Nesse caso a alteração somente se reflete na tabela do “servidor de nomes local”.
- ? A distribuição no cadastramento de informações de servidores possui a característica hierárquica, onde um determinado servidor de biblioteca digital possui em sua tabela cadastrado somente as informações (nome, endereço e escopo) de servidores que pertencem ao seu escopo e informações de um ou mais servidores que representam os escopos superiores. Essa característica hierárquica permite que o usuário possa escolher na aplicação do cliente, o escopo da busca. A partir disso a busca é “direcionada” para o servidor que atenda ao escopo do pedido, ou seja, que tenha em sua tabela servidores com o escopo do pedido do usuário. Isso também evita que um grande número de registros retorne para a aplicação do cliente, facilitando seu processo de escolha

de servidor para efetuar uma nova busca de biblioteca ou uma pesquisa por documentos digitais.

- ? A resposta de um servidor sempre volta para o cliente que originou o pedido e novos pedidos de busca são efetuados do cliente, evitando assim, um incremento nas atividades de processamento do servidor.

A seguir a alternativa escolhida será apresentada com detalhes.

4.2 Detalhamento da Busca a partir do Cliente, com Organização Hierárquica

Devido às vantagens apontadas no tópico anterior a alternativa de busca de servidores de bibliotecas digitais ou uso do serviço *Searchlib* via Busca a partir do Cliente com Organização Hierárquica se constitui naquela que melhor otimiza processo de busca de bibliotecas digitais. Ciente disso, nesse tópico são detalhados conceitos importantes apresentados anteriormente na descrição dessa alternativa.

4.2.1 Descrição do Serviço *Searchlib* – Serviço de Busca de Servidores de Bibliotecas Digitais

Searchlib é um serviço que objetiva fazer a busca de endereços de servidores de bibliotecas digitais. Nesta proposta, ele é encaminhado por um cliente Z39.50 a um servidor de biblioteca digital Z39.50 afim de solicitar os endereços de servidores de bibliotecas digitais conhecidos pelo servidor, ou seja, cadastrados na tabela do servidor.

Os parâmetros do serviço *Searchlib* são apresentados na Tabela 2 e descritos a seguir:

Nome do Parâmetro	Pedido Cliente	Resposta Servidor
Scope	Obrigatório	
ResultSearch		Obrigatório
ResultSet		Obrigatório

Tabela 2. Parâmetros do Serviço *Searchlib*

- ? **Scope**: Nesse campo da primitiva é armazenado o valor correspondente ao escopo escolhido pelo usuário na aplicação. A escolha do escopo pelo usuário é necessária para que a busca de servidores de bibliotecas digitais se

torne mais refinada. Ele é obrigatório no pedido. Os valores possíveis estão na Tabela 3. Cabe salientar que este campo armazena valores que correspondem a uma determinada forma de organização dos servidores que prevê a organização dos mesmos em uma hierarquia local, regional, nacional e global. Outras formas de organização podem ser planejadas, como por exemplo, a organização dos servidores de bibliotecas digitais por áreas temáticas, e nesse caso esse campo armazenaria valores correspondentes a essas formas.

Scope	Valor
Local	1
Regional	2
National	3
Global	4

Tabela 3. Valores Possíveis do Campo *Scope* do Serviço *Searchlib*

- ? ***ResultSearch***: Nesse campo é armazenado o valor *true* ou *false*. Ele é obrigatório na resposta do servidor. *True* indica ao cliente que o servidor Z39.50 atendeu plenamente ao pedido de *Searchlib*, ou seja, os servidores cadastrados na sua tabela possuem escopo igual ao escopo do pedido do cliente. Os registros contendo as informações (nome, endereço e escopo) desses servidores são armazenados em *ResultSet*. *False* indica que o pedido solicitado falhou, pois o servidor não é “servidor de nomes” ou é um “servidor de nomes” com o “tipo” inferior ao escopo indicado no pedido de serviço. No caso de falha, é armazenado no parâmetro *ResultSet* os registros contendo informações de “servidores de nomes” que possam atender ao escopo do pedido, devendo portanto, a nova busca ser direcionada para um desses servidores automaticamente.
- ? ***ResultSet***: Nesse campo é armazenado o conjunto composto por um ou mais registros com nome, endereço e escopo dos servidores de bibliotecas digitais. Ele é obrigatório na resposta. A Tabela 4 ilustra o formato desse campo.

Nome da Biblioteca Digital	Endereço	Escopo

Tabela 4. Formato do Campo *ResultSet* do Serviço *Searchlib*

4.2.2 Variáveis de Estado do Servidor

Todo servidor de biblioteca digital Z39.50 deve manter as seguintes variáveis de estado:

- ? **Tipo do servidor de biblioteca digital:** todo o servidor de biblioteca digital deve possuir uma classificação ou tipo que pode ser: “servidor local”, “servidor de nomes local”, “servidor de nomes regional”, “servidor de nomes nacional” ou “servidor de nomes global”. Essa classificação se dá através de organização hierárquica dos servidores em níveis ou escopos, a qual será explicitada na seção 4.2.4. O “servidor local” é um servidor de biblioteca digital simples de alguma instituição. O “servidor de nomes local” é um servidor escolhido entre os servidores locais de uma instituição, possuindo essa designação por ser o responsável por armazenar as informações (nome, endereço e escopo) de todos os “servidores locais” da instituição e as informações de alguns servidores de tipo superior. O “servidor de nomes regional” é um servidor escolhido entre os “servidores de nome local” e é responsável por armazenar as informações (nome, endereço e escopo) de todos os “servidores de nome local” de uma determinada região e as informações de alguns servidores de tipo superior. O “servidor de nomes nacional” é escolhido entre os “servidores de nomes regionais” e armazena as informações de todos servidores regionais e as informações de alguns servidores de tipo superior. O “servidor de nomes global” é escolhido entre os “servidores de nomes nacionais” e armazena as informações de todos servidores nacionais. A escolha do tipo de um servidor depende da instituição onde o servidor se encontra e também de uma instituição encarregada de gerenciar essa organização hierárquica de servidores de bibliotecas digitais. Após a escolha do tipo de servidor ser estabelecida,

isso deve ser configurado no módulo servidor do Z39.50. O estabelecimento do tipo do servidor é muito importante, pois ele sempre verifica o seu tipo no momento do recebimento de um pedido *Searchlib* e depende disso o conteúdo da resposta de *Searchlib*.

- ? **Tabela de nomes:** todos os servidores de bibliotecas digitais devem possuir uma tabela que armazena informações (nome, endereço e escopo) de outros servidores de bibliotecas digitais. A tabela de nomes é detalhada na seção que segue.

4.2.3 Tabela de Nomes

Um dos pontos chaves da solução escolhida está na distribuição hierárquica do cadastramento de nomes e endereços de servidores de bibliotecas digitais. Nesta alternativa, todo servidor de biblioteca digital, ou simplesmente biblioteca digital, deve manter uma “tabela de nomes”, cujo formato é mostrado na Fig. 8, no qual são cadastrados os dados referentes à outros servidores de bibliotecas digitais.

Nome da Biblioteca Digital	Endereço	Escopo

Figura 8. Formato da Tabela de Nomes de um Servidor de Biblioteca Digital

Os campos desta tabela são os seguintes:

- ? **Nome da Biblioteca Digital:** esse campo armazena o nome da biblioteca digital. Ele é necessário para que o nome da biblioteca digital apareça na aplicação do cliente.
- ? **Endereço:** esse campo se destina a armazenar o endereço de rede do servidor de biblioteca digital. Ele é necessário para que novas buscas de biblioteca aconteçam ou para que pesquisas por palavras-chave sejam efetuadas nesse endereço.

? **Escopo**: esse campo se destina a armazenar o escopo ou nível cada servidor de biblioteca digital cadastrado na tabela. O escopo do servidor pode ser Local, Regional, Nacional ou Global.

4.2.4 Hierarquia de Servidores

A hierarquia de servidores é uma forma de organizar o cadastramento de servidores de bibliotecas digitais nas tabelas de nome de servidores de bibliotecas digitais. Conforme apresentado inicialmente na seção 4.2.2, esta proposta sugere uma organização em quatro níveis, também chamados de escopo que são: Local, Regional, Nacional ou Global. Outros tipos de organização podem ser definidos.

ESCOPO LOCAL

A organização de servidores no escopo ou nível de hierarquia local acontece no contexto de uma determinada instituição onde podem existir vários servidores de bibliotecas digitais e entre esses um ou mais servidores são escolhidos como “**servidor de nomes local**”. Conforme explicado na seção

A Fig. 9 demonstra essa organização: No contexto de uma instituição, por exemplo UFSC, existem vários servidores de bibliotecas digitais: Biblioteca do Centro Tecnológico, Biblioteca Universitária e Biblioteca do Centro de Ciências da Saúde. Entre esses servidores um ou mais podem ser escolhidos como “servidor de nomes local”. Neste caso, foram escolhidos o servidor da Biblioteca Universitária e o servidor da Biblioteca do Centro Tecnológico.

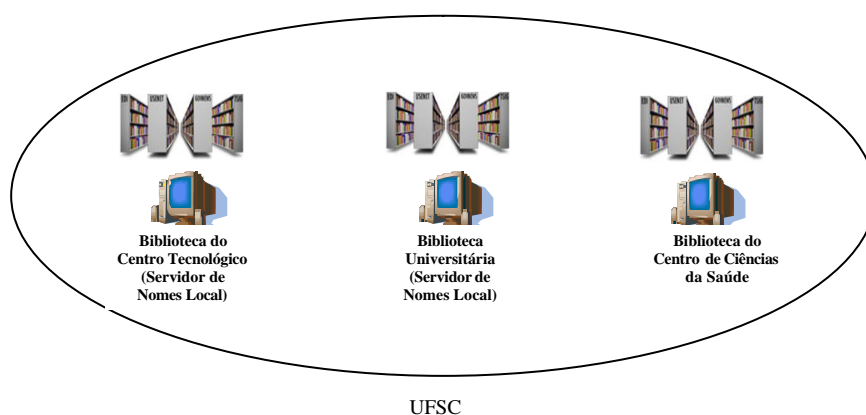


Figura 9. Escopo Local de uma Organização de Servidores de Bibliotecas Digitais

A tabela do “servidor de nomes local” contém nome, endereço e escopo (local) de todos os servidores de bibliotecas que pertencem ao escopo local dessa instituição e o nome, endereço e escopo (regional) obrigatório de um ou mais “servidores de nomes regionais”. Pode conter também, embora seja opcional, nome, endereço e escopo de um ou mais “servidores de nomes nacional e global”.

A Fig. 10 demonstra a tabela de um dos “servidores de nomes locais” da figura anterior, no caso, a Biblioteca Universitária, onde está cadastrado o nome, endereço e escopo local dos outros servidores locais da instituição, que são a Biblioteca do CTC, a Biblioteca do CCS e a própria Biblioteca Universitária que é cadastrada na sua própria tabela como servidor com escopo local, e o endereço de um “servidor de nomes regional” cujo escopo é Regional e será explicado posteriormente.

Nome da Biblioteca Digital	Endereço do Servidor	Escopo
Biblioteca X	x.x.br	Regional
Biblioteca Universitária	bu.ufsc.br	Local
Biblioteca CTC	ctc.ufsc.br	Local
Biblioteca CCS	ccs.ufsc.br	Local
...

Figura 10. Tabela de Nomes de um Servidor de Nomes Local-Biblioteca Universitária

Caso um “servidor de nomes local” receba um pedido de busca de biblioteca ou *Searchlib* com escopo local, ele retorna na resposta do serviço (*ResultSearch*) *True*, indicando que a busca foi atendida e retorna também em *ResultSet* as informações de todas as bibliotecas que possuem escopo local .

Caso o escopo de *Searchlib* for diferente de local, ele retorna na resposta do serviço (*ResultSearch*) *False* e em *ResultSet* as informações relativas aos “servidores de nomes” de escopo igual ao escopo de *Searchlib* ou de escopo imediatamente superior ao escopo local. No exemplo da Figura 10, se o escopo de *Searchlib* for diferente de local, ele retorna as informações relativas à Biblioteca X. Desta forma, o cliente pode encaminhar pedido novamente de forma automática para a Biblioteca X.

Os demais servidores de bibliotecas digitais que não são “servidor de nomes” são conhecidos como “**servidor local**”. Este tipo de servidor também possui uma tabela de nomes, na qual deve constar apenas o nome e endereço dos “servidores de nomes locais” da instituição.

A Fig. 11 ilustra a suposta tabela de nomes de um “servidor local”, ou seja, de servidor que não é “servidor de nomes” na instituição UFSC. Ela mantém necessariamente o nome, endereço e escopo dos “servidores de nomes locais” da instituição, neste caso a Biblioteca Universitária e a Biblioteca do CTC. Esta tabela deve estar armazenada em todos os servidores de bibliotecas digitais da instituição que não sejam “servidores de nome”, como por exemplo, na Biblioteca do CCS.

Nome da Biblioteca Digital	Endereço	Escopo
Biblioteca Universitária	bu.ufsc.br	Local
Biblioteca CTC	ctc.ufsc.br	Local

Figura 11. Tabela de Nomes de um Servidor Local-Biblioteca do CCS

Caso uma biblioteca que não é “servidor de nomes”, ou seja, é “servidor local” receba um pedido de busca de biblioteca ou *Searchlib*, ela retorna na resposta do serviço (*ResultSearch*) *False*, indicando que a busca não foi atendida e retorna também em *ResultSet* as informações de todos os “servidores de nomes locais” que constam em sua tabela. O cliente recebe essas informações e reencaminha o pedido automaticamente para um destes servidores. No exemplo da Figura 11, o “servidor local” retorna as informações relativas à Biblioteca Universitária e Biblioteca do CTC que são “servidores de nomes locais”. Desta forma, o cliente recebe a resposta e encaminha novamente o pedido de forma automática para um desses servidores.

A definição de mais de um “servidor de nomes local” na tabela de nome de um “servidor local” permite aumentar o nível de tolerância. Acontece à nível do cliente a escolha do “servidor de nomes” que será utilizado. Caso o acesso a um servidor falhe, o cliente pode reencaminhar o pedido para outro servidor de nomes da instituição. É de responsabilidade do cliente o controle de erro.

O campo escopo na tabela de nomes de um servidor que não seja servidor de nomes não terá nenhum efeito. Ou seja, o escopo definido na primitiva *Searchlib* será ignorado.

ESCOPO REGIONAL

A organização dos servidores no nível de hierarquia regional se dá, por exemplo, no contexto de uma determinada região que pode englobar alguns municípios, um estado ou mais de um estado. É escolhido entre os “servidores de nomes locais” um ou mais servidores que são chamados de “**servidor de nomes regional**”.

Como um “servidor de nomes regional” também é um “servidor de nomes local”, a sua tabela de nomes deve manter as informações obrigatórias apontadas no item anterior. Além disso, esta tabela de nomes mantém necessariamente o nome, endereço e escopo de todos os “servidores de nomes locais” da região. Esta tabela também deve manter obrigatoriamente o nome, endereço e escopo de um ou mais “servidores de nomes nacionais”. Opcionalmente, esta tabela de nomes pode conter também o nome, endereço e escopo de um ou mais servidores de nomes de escopo global.

A Fig. 12 demonstra essa organização: No contexto de uma região, por exemplo, Santa Catarina, existem várias instituições como UFSC, UDESC, UNISUL, etc, que possuem servidores de bibliotecas digitais, sendo que entre esses servidores um ou mais são escolhidos como “servidor de nomes local” da instituição. No escopo regional, um ou mais “servidores de nomes locais” devem ser escolhidos como “servidor de nomes regional”. Neste caso, o “servidor de nomes local” da instituição UFSC também é “servidor de nomes regional”. Para aumentar a tolerância a falhas podem-se selecionar outros servidores regionais.

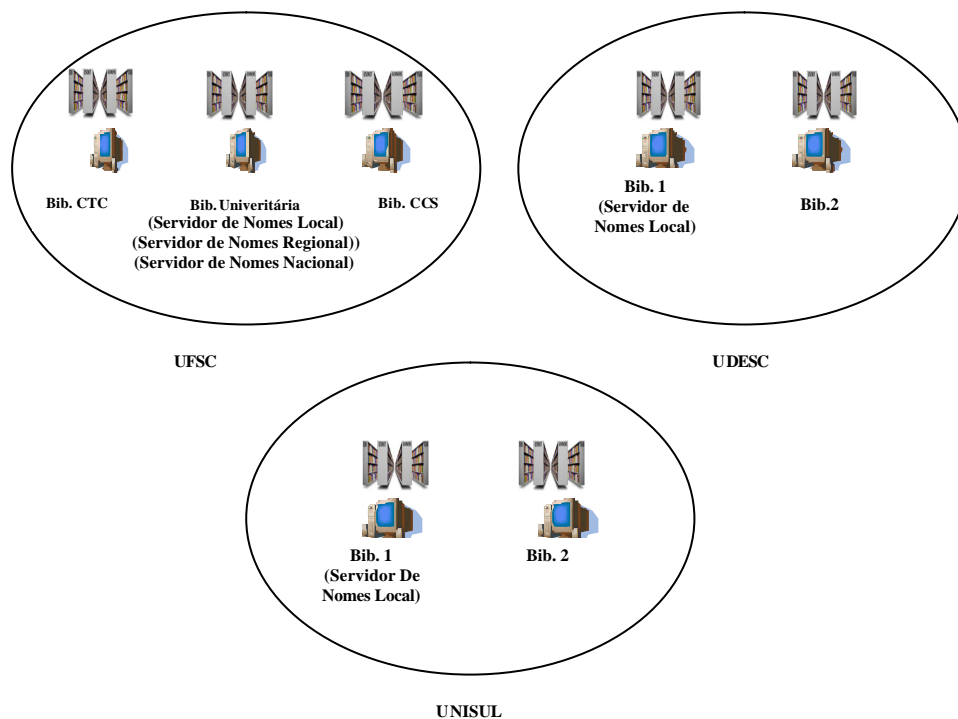


Figura 12. Escopo Regional de uma Organização de Servidores de Bibliotecas Digitais

A Fig. 13 demonstra a tabela do “servidor de nomes regional” da Fig. 12 que é a Biblioteca Universitária, onde está cadastrado o nome, endereço e escopo dos “servidores locais” da instituição, mais o nome, endereço e escopo de todos os “servidores de nomes local” que pertencem à região do “servidor de nomes regional” e o nome, endereço e escopo de um ou mais servidores de nomes nacional.

Nome da Biblioteca Digital	Endereço do Servidor	Escopo
Biblioteca X	x.x.br	Nacional
Biblioteca Universitária-UFSC	bu.ufsc.br	Regional
Biblioteca 1-UDESC	bib1.udesc.br	Regional
Biblioteca 1-UNISUL	bib1.unisul.br	Regional
...
Biblioteca Universitária	bu.ufsc.br	Local
Biblioteca CTC	ctc.ufsc.br	Local
Biblioteca CCS	ccs.ufsc.br	Local
...

Figura 13. Tabela de Nomes de um Servidor de Nomes Regional-Biblioteca Universitária

Caso uma biblioteca que é “servidor de nomes regional” receba um pedido de busca de biblioteca ou *Searchlib* com escopo local ou regional, ela retorna na resposta do serviço (*ResultSearch*) *True*, indicando que a busca foi atendida e retorna também em *ResultSet* as informações de todas as bibliotecas que possuem escopo local ou regional (conforme o escopo do pedido). Se o escopo de *Searchlib* for regional, o cliente recebe as informações dos servidores que são “servidor de nomes local”. No exemplo da Fig. 13 o cliente recebe as informações relativas à Biblioteca Universitária da UFSC, Biblioteca 1 da UDESC e Biblioteca 1 da UNISUL, que são “servidores de nomes locais”. Portanto, para chegar até um servidor local (que é o alvo de toda a pesquisa) o usuário deve “ramificar” a busca de bibliotecas, ou seja, deve escolher um ou mais “servidor de nomes local” e emitir um novo pedido de busca de bibliotecas.

Caso o escopo de *Searchlib* for diferente de local ou regional, ele retorna na resposta do serviço (*ResultSearch*) *False* e em *ResultSet* as informações relativas aos “servidores de nomes” de escopo igual ao escopo de *Searchlib* ou de escopo imediatamente superior ao escopo regional. No exemplo da Figura 13, se o escopo de *Searchlib* for nacional, ele retorna as informações relativas à Biblioteca X. Desta forma, o cliente pode encaminhar o pedido novamente de forma automática para a Biblioteca X.

ESCOPO NACIONAL

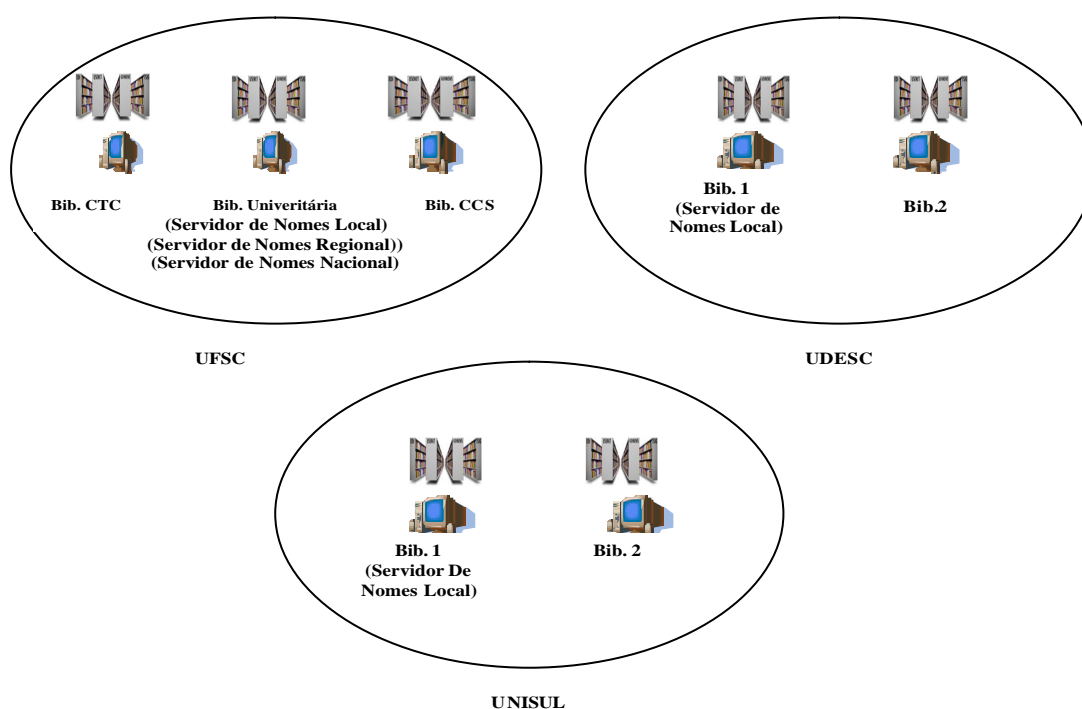
A organização dos servidores no escopo ou nível de hierarquia nacional se dá no contexto de um país. É escolhido entre os “servidores de nomes regionais” um ou mais servidores que são chamados de “**servidor de nomes nacional**”.

Como um “servidor de nomes nacional” também é um “servidor de nomes regional” e também um “servidor de nomes local”, a sua tabela de nomes deve manter as informações obrigatórias apontadas nos itens anteriores. Além disso, esta tabela de nomes mantém necessariamente o nome, endereço e escopo de todos os “servidores de nomes regionais” da nação. A tabela também deve manter obrigatoriamente o nome, endereço e escopo de um ou mais “servidores de nomes globais”.

A Fig. 14 demonstra essa organização: No contexto de um país, por exemplo, Brasil, existem várias regiões como SC, RS, etc, que por sua vez possuem várias

instituições, que também possuem vários servidores de bibliotecas digitais. Em nível de instituição (escopo local) é escolhido um ou mais servidores como “servidor de nomes local”. Em nível de região é escolhido entre os “servidores de nomes locais” um ou mais servidores como “servidor de nomes regional”. Finalmente, a nível nacional é escolhido entre os “servidores de nomes regionais” um ou mais servidores como “servidor de nomes nacional”. Neste caso, existem duas regiões, RS e SC, cada qual com seu “servidor de nomes regional” e entre esses o “servidor de nomes nacional” é o servidor da Biblioteca Universitária da UFSC.

Região SC



Caso uma biblioteca que é “servidor de nacional” receba um pedido de busca de biblioteca ou *Searchlib* com escopo local ou regional ou nacional, ela retorna na resposta do serviço (*ResultSearch*) *True*, indicando que a busca foi atendida e retorna também em *ResultSet* as informações de todas as bibliotecas que possuem escopo local, regional ou nacional (conforme o escopo do pedido). Se o escopo de *Searchlib* for global, o cliente recebe as informações dos servidores que são “servidor de nomes global”. No exemplo da Fig. 15 o cliente recebe as informações relativas à Biblioteca Universitária da UFSC e Biblioteca 1 da UFRGS, que são “servidores de nomes regionais”. Portanto, para chegar até um servidor local (que é o alvo de toda a pesquisa) o usuário deve “ramificar” a busca de bibliotecas, ou seja, deve escolher um ou mais “servidor de nomes regionais” e emitir um novo pedido de busca de bibliotecas. Ele irá chegar até os “servidores de nomes locais” e o processo se repete.

Caso o escopo de *Searchlib* for diferente de local, regional ou nacional, ele retorna na resposta do serviço (*ResultSearch*) *False* e em *ResultSet* as informações relativas aos “servidores de nomes” de escopo global. No exemplo da Fig. 15, se o escopo de *Searchlib* for global, ele retorna as informações relativas à Biblioteca do Congresso dos EUA. Desta forma, o cliente pode encaminhar o pedido novamente de forma automática para a Biblioteca do Congresso dos EUA.

ESCOPO GLOBAL

Finalmente, a hierarquia global acontece a nível mundial e o “servidor de nomes global” é escolhido entre os “servidores de nomes nacionais”.

Como um “servidor de nomes global” também é um “servidor de nomes nacional” e também um “servidor de nomes regional e local”, a sua tabela de nomes deve manter as informações obrigatórias apontadas nos itens anteriores. Além disso, esta tabela de nomes mantém necessariamente o nome, endereço e escopo de todos os “servidores de nomes nacionais”.

A Fig. 16 demonstra a tabela de nomes da Biblioteca Universitária da UFSC que já é “servidor de nomes nacional”, caso ela seja também um “servidor de nomes global”. Observe que a Biblioteca Universitária é ao mesmo tempo “servidor de nomes local, regional, nacional e global”.

Nome da Biblioteca Digital	Endereço do Servidor	Escopo
Biblioteca Universitária-BR	bu.ufsc.br	Global
Biblioteca do Congresso-EUA	lcweb.loc.gov	Global
Biblioteca Louvre-FR	bu.louvre.fr	Global
...
Biblioteca Universitária-UFSC	bu.ufsc.br	Nacional
Biblioteca 1 -UFRGS	bib1.ufrgs.br	Nacional
...
Biblioteca Universitária-UFSC	bu.ufsc.br	Regional
Biblioteca 1 – UDESC	bib1.udesc.br	Regional
Biblioteca 1- UNISUL	bib1.unisul.br	Regional
...
Biblioteca Universitária	bu.ufsc.br	Local
Biblioteca CTC	ctc.ufsc.br	Local
Biblioteca CCS	ccs.ufsc.br	Local
...

Figura 16. Tabela de Nomes de um Servidor de Nomes Global-Biblioteca Universitária da UFSC

Uma biblioteca que é “servidor de global” é capaz de atender a pedidos de busca de biblioteca ou *Searchlib* com qualquer escopo, já que tem armazenado em sua tabela de nomes todos os servidores com escopo local, regional, nacional e global.

4.2.5 Organização da Hierarquia de Servidores

Todo o controle e gerenciamento, ou seja, a organização da hierarquia de servidores deve ser realizada por alguma instituição criada em cada país especialmente para esse fim. Cabe a essa instituição estabelecer os servidores de nomes a nível regional e nacional. A escolha do(s) “servidores de nomes global” deve ser realizada em comum acordo com as instituições de cada país. A escolha do “servidor de nomes local” que acontece no contexto de uma instituição deve ser feita em comum acordo pelos administradores de cada biblioteca digital da instituição e posteriormente informada para a instituição responsável.

A instituição responsável por organizar a hierarquia de servidores deve repassar as informações necessárias para os administradores de servidores de bibliotecas digitais.

A responsabilidade pela manutenção da tabela que contém os endereços de servidores que cada servidor de biblioteca digital possui é, portanto, do administrador de cada biblioteca digital de acordo com as normas dessa instituição estabelecida para gerenciar o funcionamento da hierarquia de servidores de nomes.

4.3 Cenário de Uso

A seguir, serão descritos com detalhes os componentes envolvidos no cenário de uso do serviço *Searchlib* na Busca a partir do Cliente, com Organização Hierárquica.

4.3.1 Cliente e Servidor

O módulo cliente do Z39.50, ou simplesmente o cliente, deve estar instalado em um computador onde o usuário tenha acesso, pois é a partir desse módulo que o pedido de busca por servidores de bibliotecas digitais e também o pedido de pesquisa por documentos digitais à partir de palavras-chave é efetuado. O módulo cliente deve estar integrado com uma aplicação, a qual permite a interação do usuário e possui somente um endereço de servidor de biblioteca digital para o estabelecimento da comunicação com o mesmo. Esse endereço deve ser de um servidor local que pode ser “servidor de nomes” ou não, e é o “ponto de partida” para a busca de bibliotecas digitais.

O módulo servidor, ou simplesmente servidor, deve estar instalado em um computador que mantenha uma base de dados contendo os metadados de todos os documentos digitais da biblioteca. Ele é responsável pelo recebimento do pedido de busca por servidores de bibliotecas digitais e pelo pedido de pesquisa por documentos digitais a partir de palavras-chave. A partir do pedido ele deve fornecer a resposta contendo registros com endereços de servidores ou registros com informações de localização de documentos digitais que atendem aos requisitos da pesquisa.

4.3.2 Processo de Busca de Bibliotecas Digitais

De uma maneira detalhada, o processo de busca de servidores de bibliotecas digitais na arquitetura escolhida segue os seguintes passos:

Passo 1) o usuário escolhe na aplicação que está integrada com o módulo cliente do Z39.50, o nível ou escopo da sua busca de bibliotecas: Local, Regional, Nacional, Global, ou Nenhum e clica no botão “Busca” dando início ao

processo. A Fig. 17 ilustra esse processo. No caso de escolher a opção “Nenhum” não haverá a busca de biblioteca, e a pesquisa será feita diretamente no servidor para o qual o cliente está configurado, podendo o usuário efetuar a pesquisa por documentos digitais a partir da digitação de palavras-chave. A Fig. 18 ilustra esse processo.

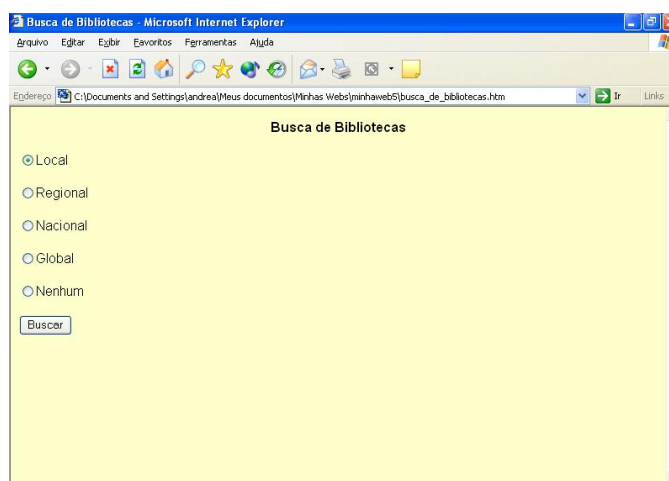


Figura 17. Interface para a Escolha do Escopo de Busca de Bibliotecas Digitais

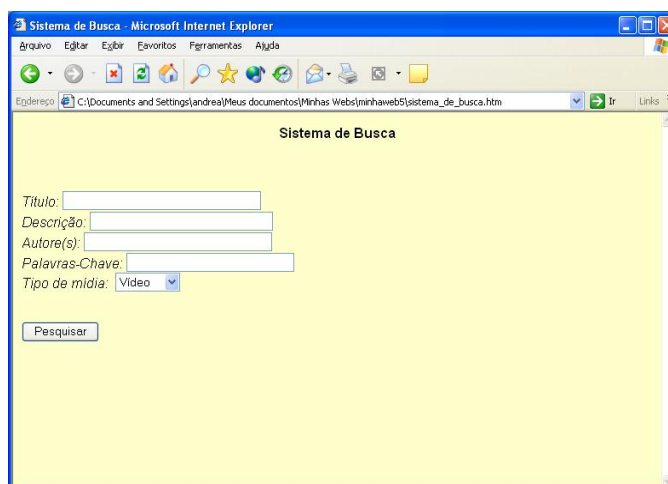


Figura 18. Interface para a Entrada de Dados Necessários para a Pesquisa de Documentos Digitais

Passo 2) cliente emite pedido de busca de biblioteca ao servidor de biblioteca que está configurado na aplicação. No campo *Scope* da primitiva de pedido de busca, deve constar o escopo do pedido escolhido pelo usuário.

Passo 3) servidor recebe o pedido e verifica seu “tipo”, ou seja, se o mesmo é servidor de nomes Local, Regional, Nacional, Global ou um Servidor Local. Essa verificação é feita através de uma variável de ambiente que deve ser configurada pelo administrador do servidor de biblioteca digital. A resposta vai depender dessa verificação:

- a) **Servidor pesquisado pelo cliente é “servidor de nomes”:** neste caso primeiramente é verificado o escopo do pedido (escolhido pelo usuário) e a partir disso, duas situações podem ocorrer:

a1) Se o escopo do pedido for inferior ou igual ao tipo do servidor pesquisado: essa situação acontece, por exemplo, quando o tipo do servidor é Regional e o escopo do pedido é Local ou o tipo do servidor é Local e o escopo do pedido é Local. Nesse caso, com certeza o servidor pesquisado possui na sua tabela registros de servidores cujo campo escopo é igual ao escopo do pedido. Esses registros são selecionados e retornados para o cliente no campo *ResultSet* da primitiva de resposta da busca. Também é retornado no campo *ResultSearch* da primitiva *True*, o que indica que a busca foi positiva, ou seja, o servidor pesquisado atendeu plenamente ao pedido de busca.

b1) Se o escopo do pedido for superior ao tipo do servidor: essa situação acontece, por exemplo, quando o tipo do servidor é Local e o escopo do pedido de busca é Nacional. Nesse caso, o mesmo verifica se existem registros de servidores cujo campo escopo é igual ao escopo do pedido e retorna esses registros no campo *ResultSet* da primitiva de resposta. Se não encontrar, é retornado o(s) registro(s) do(s) servidor(es) com escopo imediatamente superior ao tipo do servidor pesquisado, no caso do exemplo acima, um servidor cujo

campo *escopo* é *Regional*. Também é retornado no campo *ResultSearch* da primitiva *False*, o que indica que a busca foi negativa, ou seja, o servidor pesquisado não atendeu plenamente ao pedido de busca e que uma nova busca de biblioteca automática deve ser efetuada pelo cliente.

- b) Servidor pesquisado pelo cliente não é “servidor de nomes”:** neste caso o servidor pesquisado possui o tipo de “servidor local” e possui na sua tabela obrigatoriamente um ou mais registros de servidores cujo campo *escopo* é *Local*, ou seja, “servidores de nomes locais”. Esses registros são selecionados e retornados no campo *ResultSet* da primitiva de resposta da busca, juntamente com a informação *False* no campo *ResultSearch* da primitiva, o que indica que o pedido de busca foi negativo, ou seja, o servidor pesquisado não é servidor de nomes e que uma nova busca de biblioteca automática deve ser efetuada pelo cliente.

Passo 4) o cliente recebe a resposta e repassa para a aplicação. Nessa resposta existe a informação de que o pedido de busca foi *True* (positivo ou satisfeito) ou *False* (negativo ou não-satisfeito) e a partir disso, duas situações podem ocorrer:

- a) resposta da busca “False”:** significa que o servidor pesquisado não possui “tipo” igual ou superior ao *escopo* do pedido de busca de bibliotecas e, portanto, uma nova busca de bibliotecas deve ser direcionada automaticamente para o endereço do primeiro registro recebido, ou seja, para o endereço do primeiro “servidor de nomes” que está no conjunto de registros recebidos. Somente no caso do “servidor de nomes” não ser localizado e existir outro servidor de nomes no conjunto de resultados, um novo pedido de busca é efetuado para esse outro servidor. O mesmo processo se repete para o caso de não ser localizado novamente.
- b) resposta da busca “True”:** significa que o servidor pesquisado é “servidor de nomes” e possui “tipo” igual ou superior ao *escopo* do

pedido. Neste caso, é retornado na resposta o conjunto de registros contendo nome, endereço e escopo dos servidores de bibliotecas digitais pesquisados. Nessa situação não é realizada uma nova busca automática e o conjunto de registros contendo os dados de servidores de bibliotecas digitais é mostrado na aplicação do cliente. A partir disso, é preciso verificar novamente o escopo do pedido, pois dependendo dele duas outras novas situações podem acontecer:

b1) escopo do pedido de busca é “Local”: como a resposta da busca foi *True*, isso significa que a pesquisa por palavras-chaves pode proceder. É fornecida ao usuário a opção de escolher um ou mais alvos de pesquisa, no caso “servidores locais” e posteriormente a possibilidade de digitação de palavras-chave para a pesquisa de documentos digitais. A Fig. 19 e a Fig. 20 ilustram essa situação.

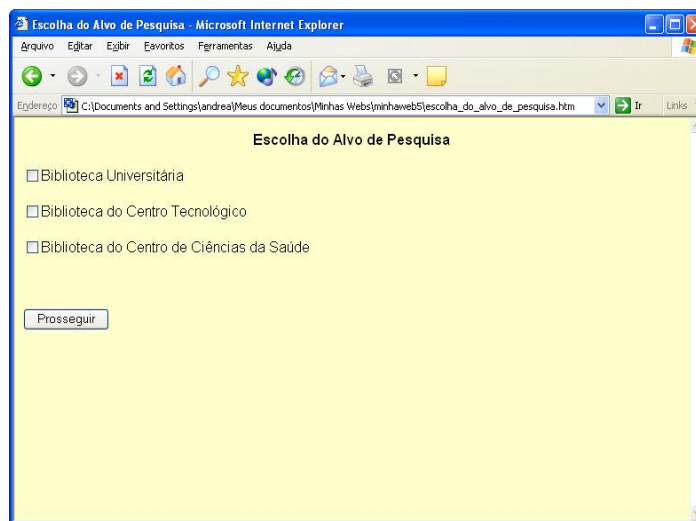


Figura 19. Interface para a Escolha do Servidor Local Alvo da Pesquisa

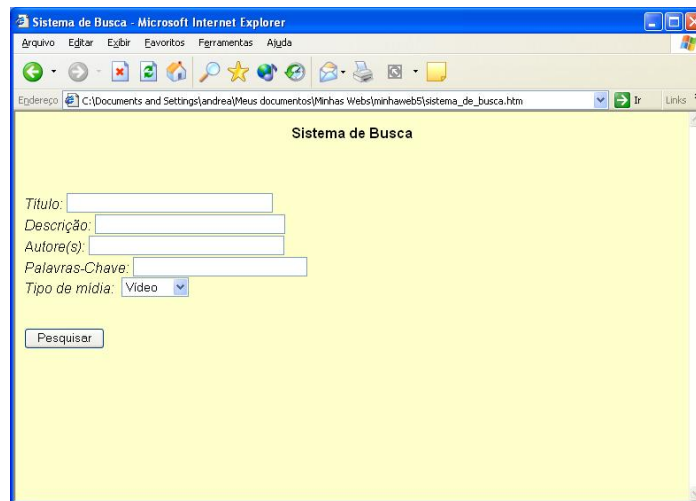


Figura 20. Interface para a Entrada de Dados Necessários para a Pesquisa de Documentos Digitais

b2) Escopo do pedido de busca não é “local”: isso significa que foi retornado a ele, por exemplo, registros de servidores cujo campo escopo é “Regional”, ou seja, registros de “servidores de nomes locais”. Portanto, o usuário pode escolher um ou vários e é emitido um novo pedido de busca de bibliotecas para cada “servidor de nomes local” escolhido. O escopo desse pedido terá sempre um nível hierárquico menor, nesse caso, “Local”. Nesse exemplo, o “servidor de nomes local” escolhido pelo usuário recebe o pedido e retorna o conjunto de registros de servidores locais. O cliente recebe e repassa para a aplicação que mostra para o usuário. Se o campo “escopo” dos registros retornados for “Local” a situação explicada no item “b1” se repete, ou seja, a interface para a digitação de palavras-chaves é mostrada ao usuário. A Fig. 21 ilustra essa situação.

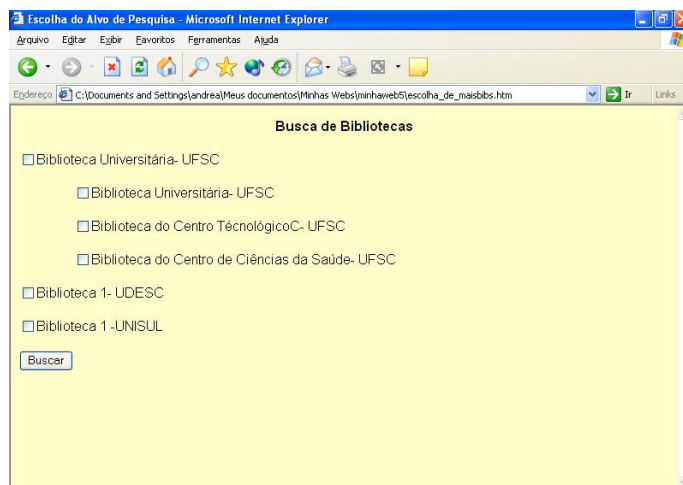


Figura 21. Interface com o Resultado da Ramificação de uma Busca de Bibliotecas Digitais

4.4 Validação da Proposta

A proposta de extensão do protocolo Z39.50 se dá através da criação de um novo serviço chamado *Searchlib* que objetiva buscar informações como nome e endereço de servidores de bibliotecas digitais. Após o estudo de algumas propostas, concluiu-se pelos motivos apresentados na seção 4.1.4 que a melhor implementação para esse novo serviço é a estrutura denominada “Busca a partir do Cliente, com Organização Hierárquica”.

Para promover uma avaliação prática do funcionamento do serviço *Searchlib* dentro dessa proposta foi desenvolvida uma aplicação que simula o funcionamento do mesmo. A aplicação foi desenvolvida na linguagem de programação Java utilizando o recurso de *threads* ou fluxos de execução. Essa opção se deu porque é possível se ter (em Java) vários *threads* executando concorrentemente (*multithreading*) e esses *threads* podem compartilhar dados em comum, tornando-se adequado, portanto, para simular a comunicação entre um cliente (*thread* Cliente) e vários servidores de bibliotecas digitais (*threads* Servidores).

A aplicação apresenta um cenário no qual constam 1 Cliente (*thread* Cliente) e 11 Servidores de Bibliotecas Digitais (*threads* Servidores), cada qual configurado com

um nome, tipo e uma tabela de nomes, ou seja, simulando o funcionamento de um servidor de biblioteca digital real.

A Fig. 22 apresenta o cenário dos servidores de bibliotecas digitais situados dentro de uma hierarquia local, regional, nacional e global (conforme a recomendação da proposta escolhida).

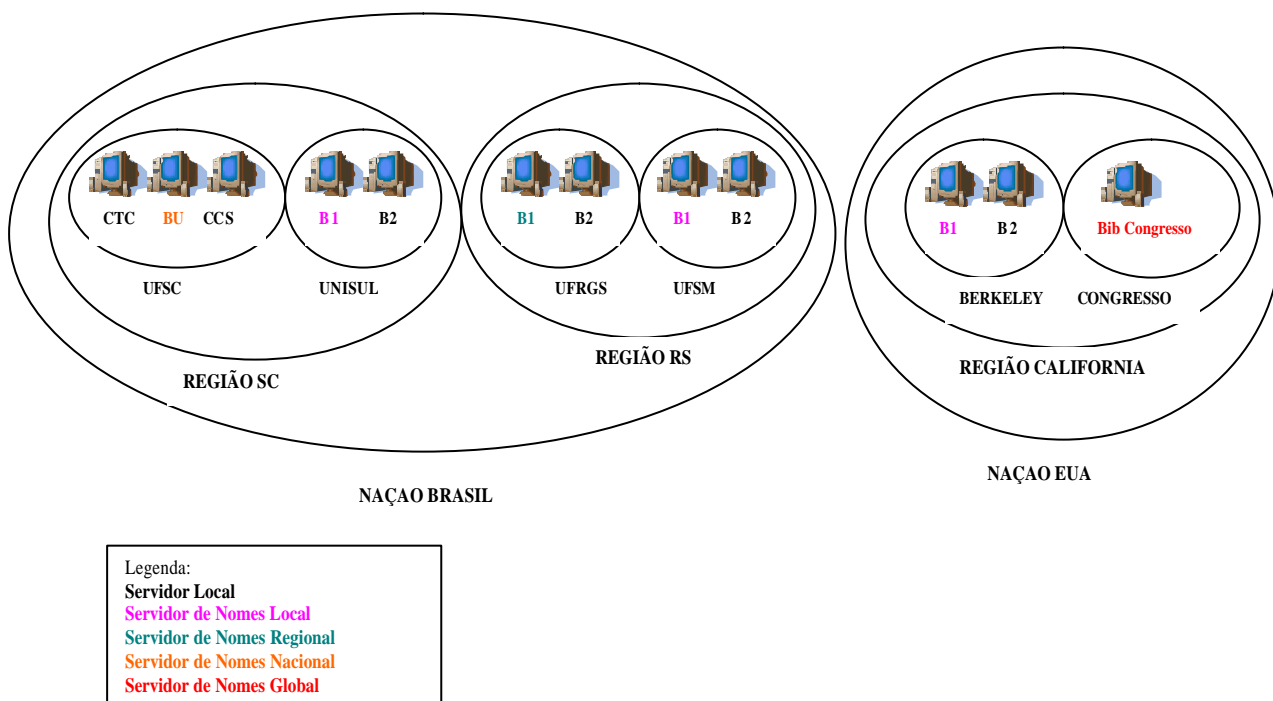


Figura 22. Cenário dos Servidores de Biblioteca Digitais Utilizados na Simulação

Para a execução da aplicação foram mapeados todos os escopos de busca (Local, Regional, Nacional ou Global) de servidores de bibliotecas digitais, ou seja, todos os tipos de pedidos de *Searchlib*, para todos os tipos de servidores de bibliotecas digitais (“servidor local”, “servidor de nomes local”, servidor de nomes regional, “servidor de nomes nacional”, “servidor de nomes global”). A Tabela 5 apresenta a relação do mapeamento realizado.

Nome do Servidor	Tipo do Servidor	Escopo da Busca	Resposta da Busca	Resposta do Servidor para Nova Busca Automática:
Biblioteca do CTC	Servidor Local	Local	Negativo	Biblioteca Universitária
Biblioteca do CTC	Servidor Local	Regional	Negativo	Biblioteca Universitária
Biblioteca do CTC	Servidor Local	Nacional	Negativo	Biblioteca Universitária
Biblioteca do CTC	Servidor Local	Global	Negativo	Biblioteca Universitária
Biblioteca do CCS	Servidor Local	Local	Negativo	Biblioteca Universitária
Biblioteca do CCS	Servidor Local	Regional	Negativo	Biblioteca Universitária
Biblioteca do CCS	Servidor Local	Nacional	Negativo	Biblioteca Universitária
Biblioteca do CCS	Servidor Local	Global	Negativo	Biblioteca Universitária
Biblioteca Universitária	Servidor de Nomes Nacional	Local	Positivo	
Biblioteca Universitária	Servidor de Nomes Nacional	Regional	Positivo	
Biblioteca Universitária	Servidor de Nomes Nacional	Nacional	Positivo	
Biblioteca Universitária	Servidor de Nomes Nacional	Global	Negativo	Biblioteca do Congresso
Bib. 1 da Unisul	Servidor de Nomes Local	Local	Positivo	
Bib. 1 da Unisul	Servidor de Nomes Local	Regional	Negativo	Biblioteca Universitária
Bib. 1 da Unisul	Servidor de Nomes Local	Nacional	Negativo	Biblioteca Universitária
Bib. 1 da Unisul	Servidor de Nomes Local	Global	Negativo	Biblioteca Universitária
Bib. 2 da Unisul	Servidor Local	Local	Negativo	Bib 1 da Unisul
Bib. 2 da Unisul	Servidor Local	Regional	Negativo	Bib 1 da Unisul
Bib. 2 da Unisul	Servidor Local	Nacional	Negativo	Bib 1 da Unisul
Bib. 2 da Unisul	Servidor Local	Global	Negativo	Bib 1 da Unisul
Bib. 1 da URGs	Servidor de Nomes Regional	Local	Positivo	
Bib. 1 da URGs	Servidor de Nomes Regional	Regional	Positivo	
Bib. 1 da URGs	Servidor de Nomes Regional	Nacional	Negativo	Biblioteca Universitária
Bib. 1 da URGs	Servidor de Nomes Regional	Global	Negativo	Biblioteca Universitária
Bib. 2 da URGs	Servidor Local	Local	Negativo	Bib. 1 da UFRGS
Bib. 2 da URGs	Servidor Local	Regional	Negativo	Bib. 1 da UFRGS
Bib. 2 da URGs	Servidor Local	Nacional	Negativo	Bib. 1 da UFRGS
Bib. 2 da URGs	Servidor Local	Global	Negativo	Bib. 1 da UFRGS
Bib. 1 da UFSM	Servidor de Nomes Local	Local	Positivo	
Bib. 1 da UFSM	Servidor de Nomes Local	Regional	Negativo	Bib.1 da UFRGS
Bib. 1 da UFSM	Servidor de Nomes Local	Nacional	Negativo	Bib.1 da UFRGS
Bib. 1 da UFSM	Servidor de Nomes Local	Global	Negativo	Bib.1 da UFRGS
Bib. 2 da UFSM	Servidor Local	Local	Negativo	Bib.1 da UFSM
Bib. 2 da UFSM	Servidor Local	Regional	Negativo	Bib.1 da UFSM
Bib. 2 da UFSM	Servidor Local	Nacional	Negativo	Bib.1 da UFSM
Bib. 2 da UFSM	Servidor Local	Global	Negativo	Bib.1 da UFSM
Bib. 1 de BERKELEY	Servidor de Nomes Local	Local	Positivo	
Bib. 1 de BERKELEY	Servidor de Nomes Local	Regional	Positivo	Biblioteca do Congresso
Bib. 1 de BERKELEY	Servidor de Nomes Local	Nacional	Negativo	Biblioteca do Congresso

Bib. 1 de BERKELEY	Servidor de Nomes Local	Global	Negativo	Biblioteca do Congresso
Bib. 2 de BERKELEY	Servidor Local	Local	Negativo	Bib. 1 de BERKELEY
Bib. 2 de BERKELEY	Servidor Local	Regional	Negativo	Bib. 1 de BERKELEY
Bib. 2 de BERKELEY	Servidor Local	Nacional	Negativo	Bib. 1 de BERKELEY
Bib. 2 de BERKELEY	Servidor Local	Global	Negativo	Bib. 1 de BERKELEY
Biblioteca do Congresso	Servidor de Nomes Global	Local	Positivo	
Biblioteca do Congresso	Servidor de Nomes Global	Regional	Positivo	
Biblioteca do Congresso	Servidor de Nomes Global	Nacional	Positivo	
Biblioteca do Congresso	Servidor de Nomes Global	Global	Positivo	

Tabela 5. Mapeamento dos Pedidos de *Searchlib* aos Servidores de Bibliotecas Digitais Utilizados na Simulação

Através da criação do cenário e do mapeamento de todos os escopos de pedidos *Searchlib* em todos os tipos de servidores foi possível realizar uma simulação eficiente, na qual foi constatado que a proposta apresentada nessa dissertação é factível.

O código fonte da aplicação desenvolvida para a simulação se encontra disponível no Anexo 2.

4.5 Conclusão

Nesse capítulo foi verificado inicialmente que o protocolo cliente/servidor Z39.50 necessita que o cliente conheça o endereço de um servidor de biblioteca digital para que possa estabelecer uma comunicação. Esse requisito, extremamente claro pela característica do protocolo que é cliente/servidor, faz com esse endereço deva ser previamente conhecido em tempo de codificação na aplicação do lado cliente do protocolo.

Entende-se que isso que concretiza numa limitação, pois é necessário conhecer todos os endereços de servidores de bibliotecas digitais e disponibilizar os mesmos via programação da aplicação, para que o usuário tenha a opção de escolher em qual servidor deseja realizar a pesquisa por documentos digitais.

Essa foi a justificativa para a proposta de extensão do protocolo Z39.50, que se dá na forma de um novo serviço chamado de *Searchlib*, o qual tem o objetivo de buscar informações (nomes e endereços) de servidores de biblioteca que estão armazenados em

outros servidores de bibliotecas digitais e disponibilizar para a aplicação cliente de forma dinâmica, ou seja, em tempo de execução.

Foram apresentadas e analisadas várias alternativas para a implementação desse novo serviço e concluiu-se que a melhor alternativa é a "Busca a partir do Cliente com Organização Hierárquica". Essa alternativa possibilita o cadastramento distribuído de informações (nome, endereço e escopo) de servidores de bibliotecas digitais em outros servidores observando uma característica hierárquica de escopos que são classificados em local, regional, nacional ou global, no qual os servidores estão inseridos.

Posteriormente foram detalhadas várias características do novo serviço *Searchlib* e da alternativa escolhida para a sua implementação.

Finalmente, foi apresentada a descrição de uma aplicação que objetivou simular o funcionamento do novo serviço *Searchlib* dentro da alternativa de implementação escolhida.

Capítulo 5 Conclusão Final

O objetivo do protocolo Z39.50 é possibilitar a busca e recuperação de informação distribuída, se valendo para isso de uma arquitetura cliente/servidor onde um cliente emite um pedido e um servidor retorna uma resposta.

O Z39.50 é um protocolo muito robusto, que se aplica à qualquer situação que exija a busca e recuperação de informação tanto local quanto remotamente. Porém um uso extensivo dele se dá no contexto das bibliotecas digitais, como um mecanismo para prover a interoperabilidade delas.

Nesse contexto, portanto, a aplicação do módulo cliente do protocolo deve conhecer previamente o endereço do servidor de biblioteca digital para que a pesquisa por documentos digitais possa ser efetuada.

Essa situação não foi identificada como a mais adequada, sendo, portanto, a motivação para a proposta dessa dissertação de mestrado, que objetiva sanar essa questão propondo uma extensão no protocolo Z39.50, através da criação de um novo serviço que se encarrega de buscar e recuperar informações (nomes e endereços) de servidores de bibliotecas digitais e disponibilizar para a aplicação exibir para o usuário.

Com isso, o nome e endereços de servidores de bibliotecas digitais não precisam ser conhecidos previamente, sendo então, construídos dinamicamente na aplicação do módulo cliente.

Foram estudadas e analisadas diversas alternativas de implementação do novo serviço de busca de bibliotecas digitais. Após a análise, foi optado pela escolha da alternativa denominada ‘Busca a partir do Cliente com Organização Hierárquica’, a qual verificou-se que era a proposta que tornava a busca de bibliotecas mais otimizada.

Essa proposta otimiza a busca de bibliotecas por vários motivos: primeiramente porque distribui o cadastramento de nomes e endereços de servidores dentro de outros servidores que estão dispostos numa organização lógica hierárquica local, regional, nacional e global, permitindo com isso que o usuário escolha na aplicação do módulo cliente, o escopo das bibliotecas que deseja buscar: local, regional, nacional, global ou

nenhum. A escolha do escopo de busca de bibliotecas propicia que um número reduzido de respostas contendo nome e endereço de servidores de bibliotecas digitais retornem para o cliente que originou a busca. Além disso, essa arquitetura propõe que um pedido de busca de biblioteca sempre ocorra do módulo cliente do protocolo, o que diminui a sobrecarga de processamento do servidor, já que o servidor também é responsável por atender os pedidos de consulta por documentos digitais.

Com o objetivo de avaliar de forma prática essa proposta de dissertação foi desenvolvida uma aplicação na linguagem de programação Java, através do uso de *multithreads*, que permitiu simular o comportamento do pedido do novo serviço *Searchlib* partindo de um cliente para um servidor e a resposta desse pedido para o cliente. Na simulação foram criados 1 *thread* para simular o Cliente e 11 *threads* para simular os servidores de bibliotecas digitais com todas as características de um servidor real como nome, tipo, tabela de nomes.

Este trabalho culminou com a avaliação prática descrita acima, na qual verificou-se que a proposta dessa dissertação é factível.

Como perspectiva de trabalhos futuros, propõe-se a implementação tanto do módulo cliente quanto do módulo servidor do protocolo Z39.50 com a adição do novo serviço *Searchlib* dentro da alternativa de implementação escolhida. Abre-se a possibilidade de estudos de outras formas de organização de servidores de bibliotecas digitais, como por exemplo, a organização dos servidores em áreas temáticas como computação, medicina, direito, etc, organização essa que possibilitaria a especialização de bibliotecas digitais em determinadas áreas como as citadas acima.

Referências Bibliográficas

- [ADL, 2001] Alexandria Digital Library. Disponível em: <http://www.alexandria.ucsb.edu/>. Acesso em: agosto de 2002.
- [ARL,1995] Association of Research Libraries. "Definition and Purposes of a Digital Library". October 23, 1995. Disponível em: <http://sunsite.berkeley.edu/ARL/definition.html>. Acesso em: julho de 2002.
- [BARKER, 1994] BARKER, Phillip. Eletronic Libraries: visions of the future. The Eletronic Library, Vol. 12, n.4. Aug. 1994.
- [BD-UNICAMP, 2002] Biblioteca Digital da Unicamp. Disponível em: <http://www.rau-tu.unicamp.br/nou-rau/sbu/>. Acesso em agosto de 2002.
- [BDM-UFSC, 2000] Biblioteca Digital Multimídia da UFSC. Disponível em: <http://www.lsd.inf.ufsc.br/bibdigital/>. Acesso em: agosto de 2002.
- [BDTD-USP, 2001] Biblioteca Digital de Teses e Dissertações da USP. Disponível em: <http://www.teses.usp.br/>. Acesso em: agosto de 2002.
- [DIENST, 2000] DIENST – Protocolo de Busca Distribuída. <http://www.cs.cornell.edu/cdlrg/dienst/DienstOverview.htm>. Acesso em: julho de 2002.
- [DLMT, 2002] Digital Library of MIT Theses. Disponível em: <http://thesis.mit.edu/>

- [GLADNEY,1994] GLADNEY, H. et Al.. Digital Library: Gross Structure and Requirements. Proc. Workshop on On-line Access to Digital Libraries, 1994. Disponível em: <http://citeseer.nj.nec.com/gladney94digital.html>. Acesso em: julho de 2002.
- [GUNTHER,1997] GUNTHER, O., VOISARD, A. Metadata in Geographic and Environmental Data Management. In: W. Klas, e A. Sheth, editors. Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data. McGraw Hill, 1997.
- [LOC, 2002] Library of Congress. Disponível em: <http://lcweb.loc.gov/>. Acesso em: agosto de 2002.
- [MARC, 2002] Machine Readable Cataloging. Disponível em: <http://lcweb.loc.gov/marc/>. Acesso em: julho de 2002.
- [MARCONDES, 2001] MARCONDES, Carlos Henrique, SAYÃO, Luiz Fernando. Projeto Técnico da Biblioteca Digital Brasileira em C&T(versão 1.1 – 27/08/2001). Disponível em: <http://www.ibict.br/bdb/>. Acesso em: julho de 2002.
- [NDL, 2002] National Digital Library. Disponível em: <http://lcweb2.loc.gov/amhome.html>. Acesso em: agosto 2002.
- [NDLTD, 2001] Networked Digital Library of Thesis and Dissertation. Disponível em: <http://adt.caul.edu.au/>. Acesso em: agosto de 2002.
- [NCSTRL, 2002] Networked Computer Science Technical Reference Library. Disponível em: <http://www.ncstrl.org/>. Acesso em: agosto de 2002.

- [PEREIRA,1995] PEREIRA, M. N. F. Bibliotecas virtuais: realidade, possibilidade ou alvo de sonho, Ciência da Informação, 1995, vol. 24, n° 24. Disp. em: <http://www.ibict.br/cionline/240195/24019512.pdf>
Acesso em: julho de 2002.
- [PISTORI, 2000] PISTORI, Jeferson. Arquitetura de Implementação de uma Biblioteca Digital Multimídia Distribuída. Dissertação de Mestrado-UFSC. Florianópolis, 2000.
- [SUMPTER, 1994] SUMPTER, R. Whitepaper on Data Management. Lawrence Livermore National Laboratory, The IEEE Metadata Workshop, 1994.
- [TENNANT,1999] TENNANT, R. Digital v. Electronic v. Virtual Libraries, 1999. Disponível em: <http://169.229.32.30/mydefinitions.html>. Acesso em: julho de 2002.
- [Z39.50, 2002] Especificação do Z39.50-1995 - Protocolo de Busca e Recuperação de Informação. Disponível em: <http://lcweb.loc.gov/z3950/agency/markup/markup.html>. Acesso em: julho de 2002.
- [Z39.50 M.A., 2001] Z39.50 Maintenance Agency – Agência de Manutenção do Z39.50. Disponível em: <http://www.loc.gov/z3950/agency/>. Acesso em: agosto de 2002.

Anexo 1 – Descrição dos Serviços do Protocolo Z39.50

1 - Initialization Facility

Initialization facility consiste de um único serviço: Init. O Init service permite o cliente estabelecer uma Z-Associação. No pedido de Init, o cliente propõe valores para parâmetros de inicialização. Na resposta de Init, o servidor responde com valores para os parâmetros de inicialização e esses valores podem diferir dos valores propostos pelo cliente que está em uso para a Z-Associação.

Se o servidor responde afirmativamente (Resulte = 'accept'), a Z-Associação é estabelecida. Se o cliente não aceitar os valores na resposta do servidor pode terminar a Z-Associação, pelo Close service (e pode tentar inicializar novamente). Se o servidor responder negativamente, o cliente pode tentar inicializar novamente.

Parâmetros do Init Service

Parameter Name	Client Request	Server Response
Version	m	m
Id/authentication	o	
Options	m	m
Preferred-message-size	m	m
Exceptional-record-size	m	m
Result		m
Implementation-id	o	o
Implementation-name	o	o
Implementation-version	o	o
User-information-field	o	o

Other-information	o	o
Reference-id	o	ia

Legenda:

m - Mandatory (obrigatório)

ø - Optional (opcional)

ia - If applicable (se aplicável)

1.1 - Version

O cliente e servidor indicam todas as versões que eles suportam. A versão comum mais alta é selecionada para uso, e é dito que está "em uso" para a Z-Associação. Se não há versão em comum, o servidor deveria indicar "reject" para o parâmetro Result.

Nota 1: número de versão mais alto que a mais alta versão conhecida deve ser ignorado.

1.2 - Id/Authentication

O cliente e servidor concordam, fora do escopo do padrão, se este parâmetro é ou não fornecido pelo cliente, e caso for, o valor. Este valor é usado para servidor determinar se o cliente está autorizado para entrar em comunicação com o mesmo.

1.3 - Options

Para cada uma das operações (representadas por "option bit") na tabela abaixo, o cliente propõe 'on' ou 'off' (significando 'in effect' ou 'not in effect' respectivamente) e o servidor responde correspondentemente para cada uma delas. A resposta determina se a operação é suportada.

Option Bit	Option	Description
0	Search	Ver Nota 1
1	Present	Ver Nota 1

2	Delete Result Set	Ver Nota 1
3	Resource Report	Ver Nota 1
4	Trigger Resource Control	Ver Nota 2
5	Resource Control	Ver Nota 3
6	Access Control	Ver Nota 3
7	Scan	Ver Nota 1
8	Sort	Ver Nota 1

Nota 1: Para cada tipo de operação (search, present, delete, resource-report, scan, sort, extended-services, duplicate detection) o cliente indica que pode escolher iniciar operações daquele tipo configurando o valor daquele tipo para 'in effect'; nesse caso, o servidor indica se está disposto a processar uma operação daquele tipo. Se o cliente propõe 'not in effect' para um tipo de operação em particular, o servidor também tem que especificar 'not in effect'.

Nota 2: O cliente pode propor submeter Trigger-resource-control requests; nesse caso, o servidor indica se aceitará Trigger-resource-control requests. Se o cliente propõe 'not in effect, ' o servidor também tem que especificar 'not in effect'. Se o servidor especifica 'in effect' para Trigger-resource-control, mas 'not in effect' para 'resource-control, ' então o cliente pode usar somente a função de Cancel de Trigger-resource-control. O servidor pode indicar indisposição para aceitar Trigger-resource-control request até mesmo se ele especificar 'in effect' para 'resource-control'. A indicação do servidor para aceitar Trigger-resource-control request não implica que o servidor fará alguma em ação como um resultado de um Trigger-resource-control request.

Nota 3: O cliente indica se permite o servidor invocar Resource-control and/or Access-control (isto é, enviar pedidos de Resource-control and/or Access-control). O servidor especifica que pode escolher (ou não) invocar Resource-control and/or Access-control. Se o servidor especifica 'not in effect' para resource-control (ou access-control) então não invocará resource-control (ou access-control) até mesmo se o cliente propôs 'em

effect'. Se o cliente propõe 'not in effect' para resource-control, e o servidor indica 'in effect' para resource-control, indicando que não está disposto a suprimir Resource-control request, e se realmente o cliente não pode aceitar Resource-control request, o cliente deveria terminar a Z-Associação. Se o cliente propõe 'not in effect' para access-control, e se requisitos de segurança no sistema servidor mandam que a segurança (outra que poderia ser provido pelo parâmetro Id/authentication) seja invocada no início de uma Z-Associação, então o servidor deveria rejeitar a Z-Associação (fixando o parâmetro Result para 'reject, ' e especificando 'in effect'.

1.4 - Preferred-message-size and Exceptional-record-size

O pedido de Init contém os valores propostos pelo cliente de Preferred-message-size and Exceptional-record-size, especificado em bytes. A resposta de Init contém o Preferred-message-size and Exceptional-record-size que o servidor vai usar; estes podem ser diferentes (e sobrepôr) dos valores propostos pelo cliente. Para o pedido e resposta, Preferred-message-size deve ser menor ou igual a Exceptional-record-size.

Exceptional-record-size é significativo durante uma operação Present, e somente no caso especial quando um único, excepcionalmente grande registro (isto é, maior que preferred-message-size) é requisitado dentro do pedido Present. Neste caso especial, Preferred-message-size pode ser anulado (para a operação Present), de forma que um único registro cujo tamanho pode ser tão grande, pode ser apresentado.

O fato que um único registro é pedido é um sinal de cliente que Preferred-message-size pode ser anulado. Assim Exceptional-record-size deve ser maior ou igual a preferred-message-size. No caso onde eles são iguais, Exceptional-record-size não tem nenhum significado (este é o modo para significar que o caso especial não aplicará durante a Z-Associação).

Se o cliente fornecer valores de zero por ambos os parâmetros, então o cliente está indicando "no preference" . Se o servidor responde com zero para ambos os parâmetros (apesar do que o cliente propôs), o servidor está indicando que o cliente deve estar preparado para aceitar arbitrariamente grandes registros e mensagens arbitrariamente grandes.

Nota que "message size", como em Preferred-message-size, etc., significa a soma dos tamanhos (em bytes) dos registros de resposta (recuperação e registros de diagnóstico) na resposta Present ou Search APDU (Application Protocol Data Unit (uma unidade de informação, transferida entre cliente e servidor cujo formato é especificado pelo protocolo Z39.50, consistindo em informação de aplicação de protocolo e possivelmente dados de usuário de aplicação). Este padrão não prescreve que os tamanhos de mensagem sejam estritamente obrigados; assim se o Preferred-message-size negociado for 32767, o servidor não precisa assegurar que o tamanho de mensagem não exceda aquele valor, mas preferivelmente que não exceda isto substancialmente. Além disso, o tamanho de mensagem é a soma dos tamanhos do registros não incluindo informação de controle de protocolo. Porém, este padrão não tenta distinguir o que é informação de controle de protocolo contra conteúdo de registro. Assim, se o servidor acha que enviou 32,767 bytes, mas o cliente acha que ele enviou 32,770 bytes (porque o servidor considerou que algo seja dados aquilo que o cliente pensou que era informação de controle de protocolo) o servidor concorda, e o cliente deve perdoar e não considerar isto um erro de protocolo. É recomendado que o cliente proponha um valor de Preferred-message-size que é menor que o limite atual que o cliente pode suportar.

1.5 - Result

O servidor indica se aceita ou não a Z-Associação especificando um valor de 'accept' ou 'reject' no parâmetro Result. (Se 'reject' é indicado, o cliente pode enviar outro Init request.).

1.6 - Implementation-id, Implementation-name, and Implementation-version

O pedido(request) ou resposta(response) podem opcionalmente incluir quaisquer destes três parâmetros. Eles são, respectivamente, um identificador (único dentro do cliente ou sistema de servidor), nome descritivo, e versão descritiva, para o cliente ou implementação de servidor. Estes três parâmetros de implementação são providos para a conveniência de implementadores, com a finalidade de distinguir implementações.

1.7 - User-information-field

Este parâmetro pode ser usado pelo cliente ou servidor para informação adicional não especificada por este padrão.

1.8 - Other-information

Este parâmetro pode ser usado pelo cliente ou servidor para informação adicional não especificada pelo padrão. Este parâmetro só pode ser usado se versão 3 estiver em uso.

1.9 - Reference-id

Ver tópico "Operações e Reference-id".

2- Search Facility

Search facility consiste de um único serviço: Search. O serviço de Search permite um cliente especificar a consulta a ser aplicada no bancos de dados de um sistema servidor, e receber informação sobre os resultados da consulta.

O pedido de Search permite que o cliente peça ao servidor que o mesmo aplique uma consulta à um especificado conjunto de bancos de dados no servidor e identifique registros com as propriedades indicadas pela consulta. O servidor cria um conjunto de resultados, que representa o conjunto de registros identificados pela consulta. O conjunto de resultados é um conjunto ordenado; um registro identificado pela entrada no conjunto de resultados é referenciado pela posição da entrada dentro do conjunto de resultados (começando com 1). O servidor mantém o conjunto de resultados para subseqüentes pedidos de recuperação.

Dependendo dos parâmetros da procura, um ou mais registros identificados pelo conjunto de resultados podem ser imediatamente recuperados como parte da resposta(response) de Search (a isto se refere informalmente como "piggybacking").

Recuperação é em geral uma função do serviço Present (Retrieval Facility), mas em casos especiais pode fazer parte da operação de Search.

Parâmetros do Search Service

Parameter Name	Client Request	Server Response
Query-type	m	
Query	m	
Database-names	m	
Result-set-name	m	
Replace-indicator	m	
Small-set-element-set-names	o	
Medium-set-element-set-names	o	
Preferred-record-syntax	o	
Small-set-upper-bound	m	
Large-set-lower-bound	m	
Medium-set-present-number	m	
Response-records		ia
Result-count		m
Number-of-records-returned		m
Next-result-set-position		m
Search-status		m
Result-set-status		ia
Present-status		ia
Additional-search-information	o	o
Other-information	o	o

2.1 - Query-type and Query

O parâmetro Query-type identifica o tipo de consulta, isto é, a sintaxe dos parâmetros de consulta. Seis tipos estão definidos:

- Type-0: só pode ser usado quando o cliente e servidor tiverem um acordo à priori fora do padrão.

- Tipo-1: é a consulta RPN (Notação Polonesa Inversa - assim chamada porque RPN às vezes é usada como uma representação abstrata).

- Tipo-2: é o tipo de consulta ISO8777, especificada em ISO 8777.

- Tipo-100: é o tipo de consulta Z39.58, especificada em ANSI Z39.58.

- Tipo-101: é a consulta RPN estendida (ERPN); uma extensão para a consulta Tipo-1 que permite proximidade de procura e restrição do conjunto de resultados através de atributos.

Nota: A consulta tipo-101 é idêntica para a consulta tipo-1 com a seguinte exceção: Para tipo-1, proximidade e restrição só são válidas quando versão 3 está em uso. Para tipo-101, proximidade e restrição são ambas válidas para versão 3 e versão 2. (A definição de consulta tipo-101 é independente de versão.)

- Tipo-102: é a consulta de Lista Enfileirada, definida em uma versão posterior deste padrão.

- Tipo-104: é usado para identificar consultas externamente definidas.

2.2 - Database-names

O cliente indica o conjunto de bancos de dados para os quais a consulta será aplicada.

Notas:

1 - Combinações de banco de dados: O servidor designa (através da Explain facility ou de algum mecanismo fora do padrão) que bancos de dados podem ser especificados em um pedido de Search, e em que combinações eles podem ser especificados.

Por exemplo, um servidor poderia especificar que bancos de dados A, B, e C, podem ser procurados individualmente, e que A e B podem ser procurados em combinação (mas não A e C, nem B e C).

2 - Pesquisa a Multi-Banco de Dados: Z39.50 permite mas não requer suporte para, procura multi-banco de dados em um único pedido de Search. Um cliente simplesmente pode escolher não enviar pedidos de Search que listam mais que um único banco de dados. Um servidor, em recebimento de um pedido de Search em multi-banco de dados pode fracassar a procura (diagnóstico 111: " Muitos bancos de dados especificados"; com um addinfo "maximum value " 1 fornecido). Um servidor que escolhe suportar um pedido de procura de multi-banco de dados, pode limitar tal suporte para qualquer combinação específica que ele escolhe. Por exemplo suponha um servidor provê 100 bancos de dados, nomeou "1", "2 ",..., "100 ". Ele pode declarar que cada banco de dados pode ser procurado isoladamente, e que os bancos de dados " 1 " e " 2 " podem ser procurados junto. (Ele pode declarar isto via Explain, ou, mais dinamicamente, quando recebe uma Search com uma não suportável combinação de bancos de dados pode falhar a Search com o diagnóstico 23: " Combinação especificada de bancos de dados não suportada ").

Um servidor pode definir bancos de dados virtuais que correspondem a combinações suportadas em lugar de suportar pedidos de Search à múltiplos banco de dados. Por exemplo, suponha que há 3 bancos de dados reais (A, B, e C) para qual um servidor permitirá acesso em todas as possíveis combinações; o servidor poderia expor 7 bancos de dados virtuais (A, B, C, AB, AC, AC, e ABC), assim provendo todas as possíveis combinações mas permitindo só um único (virtual) banco de dados ser especificado em um pedido de Procura.

3 - Nenhum Banco de dados Default: Não há nenhum conceito definido em Z39.50 de um "default" banco de dados. Se um servidor considera que um banco de dados particular seja o banco de dados default, poderia expressar isso através de Explain

(usando o campo “descrição” de DatabaseInfo), porém estas informações só seriam para consumo humano.

4. Case Insensitive: Cada nome de banco de dados especificado pelo servidor é um string de caracteres, e a string é case-insensitive. Assim para qualquer caráter que é uma letra, o cliente pode usar maiúscula ou minúscula, apesar de como o servidor especificou o nome.

2.3 - Result-set-name and Replace-indicator

O parâmetro Result-set-name especifica um nome a ser dado ao conjunto de resultados (criado por uma consulta-query) de forma que isto pode ser subseqüentemente referenciado (dentro da mesma Z-Associação).

Se um conjunto de resultados com o mesmo nome já existir no servidor, a ação tomada depende do valor do parâmetro Replace-indicator, como segue:

Se o valor de Replace-indicator é "on", então depois de processar a query, o conjunto de resultados existente cujo nome é especificado pelo parâmetro Result-set-name será deletado e um novo conjunto de resultados com aquele nome é criado. Se a query não pode ser processada, o conteúdo do conjunto de resultados será vazio.

Se o valor de Replace-indicator é "off", a procura não é processada, um diagnóstico de erro é devolvido pelo servidor, e o conjunto de resultados existente cujo nome é especificado pelo parâmetro Result-set-name permanece inalterado. Se um conjunto de resultados não existe com o nome especificado pelo parâmetro Result-set-name, então um conjunto de resultados é criado pelo servidor, e o parâmetro Replace-indicator é ignorado.

O conteúdo inicial do conjunto de resultados está vazio. Se nenhum registro é encontrado pela query, o conjunto de resultados permanece vazio.

Em geral, um servidor não necessita suportar, a nomeação do conjunto de resultados pelo cliente (veja Notas abaixo.).

Porém, um servidor deve suportar ao menos o conjunto de resultados cujo nome é "default". Se o cliente especifica " default " como Result-set-name, então Replace-indicator deve ser ' on'.

Um conjunto de resultados criado por um pedido de Search (isto é, especificado pelo parâmetro Result-set-name) pode ser referenciado em um pedido Present subsequente ou como um operando em uma Search subsequente. Se um conjunto de resultados nomeado " default" é criado, ele permanece disponível para referência a partir da hora que é criado até o fim da Z-Associação durante a qual é criado, ou até que:

-- Outro conjunto de resultados default seja criado, porque o nome " default" é especificado como Result-setname em um pedido de Search subsequente.

- Ele seja unilateralmente deletado pelo servidor

Qualquer conjunto de resultados diferente do conjunto de resultados nomeado "default" permanece disponível para referência a partir da hora que é criado até que o momento que for apagado através dos seguintes modos:

- Através de uma operação Delete.

- Implicitamente, porque um conjunto de resultados foi especificado com o mesmo nome em um pedido de Search, e o valor do parâmetro Replace-indicator era ' on'.

- Unilateralmente pelo servidor (a qualquer hora).

- Através do término da Z-Associação.

2.4 - Small-set-element-set-names and Medium-set-element-set-names

Estes parâmetros descrevem a composição preferida dos registros esperada na resposta de Search. Se a query resulta em um pequeno-conjunto então Small-set-element-set-names é configurado. Se a questão resultar em um médio-conjunto, então Small-set-element-set-names é configurado.

2.5 - Preferred-record-syntax

O cliente pode especificar uma sintaxe de registro preferida para a recuperação de registros.

2.6 - Small-set-upper-bound, Large-set-lower-bound, and Medium-set-presentnumber

O conjunto de resultados é considerado um "small-set", "medium-set", ou "large-set", dependendo dos valores dos parâmetros do pedido de Search Small-set-upper-bound and Large-set-lower-bound e da resposta de Search Result-count. O conjunto de resultados é um "small-set" se Result-count não é maior que small-set-upper-bound. O conjunto de resultados é um "large-set" se Result-count é maior ou igual a Large-set-lower-bound. Caso contrário, o conjunto de resultados é um medium-set. Se a query resulta em um "small-set", registros de resposta correspondendo a todos os registros de banco de dados identificados pelo conjunto de resultados são retornados ao cliente (sujeito a possíveis restrições de tamanho de mensagem). Se a query resulta em um "large-set", nenhum registro de resposta é retornado. Se a query resulta em um "medium-set", o número máximo de registros de resposta a ser retornado é especificado por Medium-set-present-number.

2.7 - Response-records

O servidor processa a query, enquanto cria um conjunto de resultados que identifica um conjunto de registros de banco de dados. Não pode ser assumido entretanto, que o processamento da query requer acesso físico aos registros do banco de dados. Um registro de banco de dados em particular poderia não ser acessível, mas esta circunstância poderia não ser reconhecida até uma tentativa ser feita para acessar o registro com a finalidade de formar uma recuperação de registro.

Depois de processar a query, o servidor tenta criar registros recuperados para serem incluídos na resposta de Search, correspondendo aos primeiros N registros de banco de dados identificados pelo conjunto de resultados (N depende dos parâmetros de pedido e Result-count). Para cada registro de banco de dados para o qual um registro de recuperação não pode ser incluído, um registro de diagnóstico substituto é substituído.

O termo registro de resposta refere-se a um registro de recuperação ou um registro de diagnóstico substituto. O parâmetro Response-records é um dos seguintes:

- N registros de resposta.
- Um número de registros de resposta, que é menos que N por causa de restrições de tamanho de mensagem.
- Um ou mais registro de diagnóstico substituto indicando que a procura não pode ser processada e por que não pode ser processada.
- Um ou mais registros de diagnóstico substituto indicando que registros não podem ser apresentados, e por que não, por exemplo "não foi fixado nome válido para banco de dados".

Nota: Se versão 2 está em uso, o servidor devolve um único registro diagnóstico substituto. Se versão 3 está em uso, o servidor devolve um ou mais registros diagnósticos substitutos.

A ordem de ocorrência de registros de resposta dentro do parâmetro Response-records está de acordo com a ordem na qual eles são identificados pelo conjunto de resultados. Cada um pode opcionalmente ser acompanhado pelo nome do banco de dados no qual o registro reside. Porém, o nome de banco de dados deve acompanhar o primeiro registro de resposta devolvido, e deve acompanhar qualquer registro de um banco de dados diferente de seu antecessor imediato. O nome de banco de dados junto a um registro não precise ser um dos nomes de banco de dados que foram incluídos no pedido de procura original (o pedido que causou a criação do conjunto de resultados a partir do qual o registro está sendo apresentado).

Quando o servidor não suporta Response Records em uma resposta de Search

Se um servidor não suporta “piggybacking”, isto é, registros de resposta em uma Search, e quando os valores de parâmetro para Small-Set-Upper-Bound and Large-Set-Lower-Bound de um pedido de Search e os valores de parâmetro para Result-Count de uma resposta de Search são tais que é esperado que a resposta inclua um ou mais

registros (por exemplo, se uma procura resulta em 5 registros e `small-set-upper-bound` no pedido de procura era 10, do servidor é esperado que ele tente devolver todos os cinco registros), o servidor deveria responder com um `Search-status` de 'success' e um `Present-status` de 'failure', e incluir um registro de diagnóstico não-substituto, por exemplo: Diagnóstico geral "1005: Resposta de registros em resposta de Search não suportada", ou "1006:Resposta de registros em resposta de Search impossível para banco de dados especificado ou combinação de banco de dados".

Note que Diagnóstico 1006 é planejado para o caso de um provedor intermediário acessar múltiplos servidores, alguns dos quais podem suportar piggybacking e alguns não, e é para o intermediário usar no caso de um servidor particular final não suportar piggybacking, para distinção do caso onde o próprio intermediário não suporta piggybacking.

Quando Query não é suportada por um Banco de dados.

Quando a query não é suportada para um dos bancos de dados (isto é, quando há pelo menos um banco de dados listado pelo Cliente no parâmetro `Database-names`, para o qual a query não é suportada, embora a query possa ser suportada para outros bancos de dados listados), o servidor deve configurar `'search-status'` para "failure", e fornecer um diagnóstico apropriado. (Por exemplo, quando um atributo não é suportado para um banco de dados, fornecer diagnóstico 1056: "Atributo não suportado para banco de dados ". Múltiplas instâncias de diagnóstico 1056 podem ser fornecidas para combinações `attribute/database`).

Quando uma query é suportada para um ou mais, mas não todos os bancos de dados, e se os servidores querem fazer o conjunto de resultados disponível ao cliente (o conjunto de resultados produzido através da aplicação da query para o subconjunto de bancos de dados suportados) o servidor pode configurar `Search-status` como "failure" e `Result-set-status` para "subset: partial, valid results available". O raciocínio para este prescrito comportamento é como segue: Configurar `Search-status` para 'success' daria ao cliente a falsa impressão que a procura foi executada por todos os bancos de dados. Para aqueles bancos de dados para os quais a query não é suportada, o cliente seria levado a acreditar que a query é realmente suportada e que nenhum registro foi achado.

Mas enquanto um status de ' success' está enganando quando parte da procura falha, ' failure', também está enganando quando parte da procura tem sucesso. Assim o parâmetro Result-set-status ajuda a destornar âmbiguo um estado de 'failure'. Além disso, Result-set-status só podem ser fornecidos quando Search-status estiver em ' failure'. Assim, configurar o status para ' failure' é prescrito, não por causa de qualquer semântica incluída de um estado de falha mas (mais pragmatically) porque proporciona para o cliente acesso para um conjunto de resultados.

2.8 - Result-count and Number-of-records-returned

O parâmetro Result-count é o número de registros de banco de dados identificados pelo conjunto de resultados. Se o conjunto de resultados está vazio, Result-count é zero.

Um valor negativo para resultCount em uma resposta de Search é inválido e constitui uma violação de protocolo. Um cliente que recebe um valor negativo de Result-count pode emitir um pedido de Close Close-reason "protocolo error"(erro de protocolo).

O parâmetro Number-of-records-returned é o número total de registros devolvidos na resposta de Search, incluindo registros de diagnóstico, substituto ou não-substituto.

2.9 - Next-result-set-position

O parâmetro Next-result-set-position assume o valor M+1, onde M é a posição do item do conjunto de resultados que identifica o registro de banco de dados que corresponde ao último registro de resposta aqueles devolvidos; ou zero se M = Result-count.

2.10 - Search-status

O parâmetro Search-status, devolvido na resposta, assume um dos seguintes valores:

Success A procura foi completada com sucesso.

Failure A procura não foi completada com sucesso.

Um valor de ' success' não implica que os registros de resposta esperados estão sendo devolvidos como parte da resposta (ver Present-status). Note também, um valor de ' success' não implica que algum registro de banco de dados foi localizado pela Search. Um valor de ' failure' implica que nenhum dos registros de resposta esperados estão sendo retornados. No caso posterior, o servidor devolve um ou mais registros de diagnóstico não-substituto(veja nota) indicando que a procura não pode ser processada.

Nota: Se versão 2 está em uso, o servidor devolve para um único registro de diagnóstico não-substituto. Se versão 3 está em uso, o servidor devolve um ou mais non-substituto registros diagnóstico.

2.11 - Result-set-status and Present-status

Estes são descritores de estado necessários para distinguir situações ambíguas que podem potencialmente acontecer durante as operações de Search e Present.

Result-set-status acontece se e somente se o valor de Search-status é ' failure' e seu valor é um dos seguintes:

Subset Parcial, valid results available.

Interim Partial results available, not necessarily valid.

None Nenhum conjunto de resultados.

Present-status acontece se e somente se o valor de Search-status é ' success' e seu valor é um dos seguintes:

Success Todos os registros de resposta esperados estão disponíveis.

Partial-1 Nem todos os registros de resposta esperados podem ser devolvidos porque o pedido foi terminado por acesso controle (access control).

Partial-2 Nem todos os registros de resposta esperados podem ser devolvidos porque eles não ajustarão dentro do tamanho de mensagem preferido.

Partial-3 Não todos os registros de resposta esperados podem ser devolvidos porque o pedido foi terminado por controle de recurso(resource controle), a pedido do cliente.

Partial-4 Nem todos os registros de resposta esperados podem ser devolvidos porque o pedido foi terminado por controle de recurso(resource control), pelo servidor.

Failure Nenhum dos registros de resposta esperados podem ser devolvidos.

Um ou mais registros de diagnóstico não-substituto são devolvidos.

2.12 - Additional-search-information

Na resposta o servidor pode usar este parâmetro para transportar informação, que é um by-product do processo de procura, incluindo, por exemplo, contas de resultado intermediárias, por que registros particulares foram devolvidos, ou se um atributo particular foi usado para procurar um banco de dados. No pedido o cliente pode usar este parâmetro para indicar o formato preferido ou conteúdo daquela informação. Format de informações de usuário SearchResponse-1 está definido em Apêndice USR. Este parâmetro só pode ser usado quando versão 3 está em uso.

2.13 - Other-information

Este parâmetro pode ser usado pelo cliente ou servidor para informação adicional não especificada pelo padrão. Este parâmetro só pode ser usado quando versão 3 está em uso.

3 – Retrieval Facility

Retrieval Facility consiste em dois serviços: Present e Segment.

O cliente envia um pedido Present para pedir registros de resposta de acordo com uma posição dentro de um conjunto de resultados mantido pelo destino. O servidor responde enviando uma resposta Present, contendo os registros de resposta pedidos.

Alternativamente, se segmentação está em uso e os registros de resposta pedidos não se ajustaram dentro da mensagem de resposta Present, o servidor pode segmentar a resposta enviando um ou mais pedidos Segment antes da resposta Present.

O pedido Segment (se houver) junto com a resposta Present são referenciados como resposta Present agregada. Cada pedido Segment como também a resposta Present é referenciada como um segmento da resposta Present. Se uma resposta Present agregada consiste em um único segmento (isto é, somente uma resposta Present) é chamado uma resposta Present simples.

4 - Result-set-delete Facility

Result-set-delete facility consiste de um único serviço: Delete.

O Delete service permite um cliente pedir que o servidor apague conjuntos de resultado especificados, ou todos os conjuntos de resultado, criados durante a Z- Associação. O servidor responde através do relato de informações pertencentes ao resultado da operação.

5 - Access Control Facility

Access Control facility consiste de um único serviço: Access-control.

O serviço de Access-control permite um servidor recusar um cliente. A recusa poderia pertencer a uma operação específica ou a uma Z- Associação. O mecanismo de request/response de Access-control pode ser usado para suportar recusas de controle de acesso ou autenticação, inclusive recusas de senha, chave pública de sistemas criptográficos e autenticação algorítmica.

Um cliente deve estar preparado para aceitar e responder um pedido de Access-control do servidor se Access-control estiver em uso. Um servidor pode emitir um pedido de Access-control que é parte de uma específica (ativa) operação, ou que pertence à uma Z- Associação.

Se operações simultâneas estão em uso:

Se o pedido de Access-control inclui um Reference-id: O Reference-id fornecido tem que corresponder a uma operação ativa; o pedido de Access-control é parte daquela operação. A resposta de Access-control também tem que incluir aquele Reference-id.

Se o pedido de Access-control não inclui um Reference-id: O pedido e resposta de Access-control não são parte de qualquer operação, eles pertencem à Z-Associação.

Se operações consecutivas estão em uso: O servidor só pode emitir um pedido de Access-control quando houver uma operação ativa; o pedido de Access-control e subsequente resposta são parte daquela operação e têm que incluir o Reference-id da operação (que é assumido ' null' se não presente no pedido inicial).

6 -Accounting/Resource Control Facility

Accounting/Resource Control facility consiste em três serviços:

- Resource-control service, invocado pelo servidor, ou como parte de uma operação ativa (de qualquer tipo) ou pertencendo à Z-Associação;
- Trigger-resource-control service, invocado pelo cliente como parte de uma operação ativa (de qualquer tipo menos Init);
- Resource-report service, invocado pelo cliente para iniciar uma operação de Resource-report.

O serviço de Resource-control permite ao servidor enviar um pedido de Resource-control, que poderia incluir um Resource report. O report poderia notificar o cliente que o consumo de recursos previstos ou atual excederá os limites, e pede o consentimento do cliente para continuar uma operação, pela resposta de Resource-control. O servidor poderia, por exemplo, informar o cliente sobre o estado atual de um conjunto de resultados sendo gerado no servidor durante uma operação de Search, e indicar informações sobre o progresso da operação.

O serviço de Trigger-resource-control permite ao cliente pedir que o servidor inicie o serviço de Resource-control, ou cancele a operação.

O serviço de Resource-report permite ao cliente pedir que o servidor envie um Resource-report pertencente a uma operação completada ou a uma Z-Associação.

7 - Sort Facility

Sort facility consiste de um único serviço: Sort.

O Sort service permite um cliente pedir que o servidor ordene um conjunto de resultados (ou faça a união de múltiplos conjuntos de resultados e então ordene). O cliente especifica a seqüência de ordenação de elementos. O conjunto de resultados será ordenado de acordo com a seqüência especificada, e subseqüentes pedidos de conjuntos de resultado novamente serão construídos pelo servidor de acordo com essa ordem.

8 – Browse Facility

Browse facility consiste de um único serviço: Scan.

O Scan service é usado para examinar uma ordenada lista de termos (termos de assunto, nomes, títulos, etc.). A ordenação da lista de termos é definida pelo servidor. O cliente especifica uma lista de termos para examinar e um termo iniciante (implicitamente, especificando uma combinação de attribute/term e um id de banco de dados), o tamanho dos passos de exame, o número desejado de entradas e posição do termo iniciante na resposta.

9 – Extended Facility

Extended facility consiste de um único serviço: Extended-services.

O serviço Extended-services (ES) permite um cliente criar, modificar ou deletar um pacote de tarefas no servidor. O servidor mantém pacotes de tarefas em um banco de dados especial, descrito na seção 3.2.9.2. Um pacote de tarefas pertence a uma tarefa de ES.

Um serviço estendido é um tipo de tarefa, relacionado a recuperação de informação, mas não definido como um serviço Z39.50. A execução de uma tarefa pelo

servidor está fora do escopo de Z39.50. Os serviços estendidos definidos por este padrão em seção 3.2.9.1.2. Definições desses serviços são incluídos no Apêndice 8 EXT.

O cliente envia um pedido de ES para o servidor pedindo a execução de uma tarefa. O pedido inclui parâmetros que o servidor usa para construir o pacote de tarefas. O servidor confere o pedido para validade, para consistência com os privilégios de acesso do usuário e possivelmente para outras limitações dependentes do servidor. O servidor envia uma resposta de ES indicando que o pedido foi aceito ou fornece uma indicação da razão pela qual o pedido foi rejeitado.

O serviço de ES é um serviço confirmado, iniciado pelo cliente. A operação de ES consiste em um pedido do cliente e uma resposta do servidor, possivelmente com intervenientes mensagens de Access-control ou Resource-control. Porém, embora o pedido pode resultar na iniciação de uma tarefa, a tarefa não é considerada parte de uma operação de ES Z39.50. O servidor responde, o qual completa a operação de ES, necessariamente não sinaliza conclusão da tarefa. Uma tarefa pode ter um tempo de vida que excede uma única Z-Associação.

Execução da operação de ES resulta na criação de um pacote de tarefas, representada por um registro de banco de dados no banco de dados de ES.

Por exemplo, quando um servidor cria um pacote de tarefas de tipo PersistentResultSet, um (persistente) conjunto de resultados é criado, representado pelo pacote de tarefas criado, na forma de um registro no banco de dados de serviços estendidos. Quando aquele pacote é recuperado subsequente por um cliente, ou na mesma ou em uma Z-Associação diferente, uma cópia daquele conjunto de resultados persistente é colocada disponível àquela Z-Associação, como um conjunto de resultados Z39.50 (isto é, como um conjunto de resultados passageiro; um nome de conjunto de resultados para uso durante a Z-Associação, é incluído dentro do pacote de tarefas). Quando um cliente apaga o pacote de tarefas, o conjunto de resultados persistente é apagado.

Um pacote de tarefas contém parâmetros, alguns dos quais são comuns a todos os pacotes de tarefa sem levar em consideração o tipo de pacote, e outros que são específicos a um serviço estendido particular. Entre os parâmetros comuns (indicados na tabela abaixo, listados abaixo de "task package parameter" na coluna direita), alguns são fornecidos pelo cliente como parâmetros no pedido ES, e são usados pelo servidor para formar o pacote de tarefas; alguns desses fornecidos pelo cliente podem ser anulados (sobrescritos) pelo servidor. Outros são fornecidos pelo servidor. Os parâmetros específicos são derivados dos parâmetros Task-specific-parameters do pedido ES.

10 – Explain Facility

A Explain facility permite um cliente obter detalhes de implementação de um servidor, inclusive bancos de dados disponíveis por pesquisa, conjuntos de atributos e conjuntos de diagnósticos usados pelo servidor, schema, sintaxe de registros e definições de especificação de elementos suportadas para recuperação. Servidores que suportam a Explain facility:

- proveem acesso (pelos serviços de Search e Present de Z39.50) para um banco de dados com o nome IR-Explain-1 (chamado de " Explain database");
- suportam o conjunto de atributos explain, exp-1, definido em Apêndice 3 ATR (que define um conjunto de Use atributos e importam bib-1 non-use atributos); e
- suportam a Explain sintaxe que está definido e registrado em Apêndice 5 REC.

Um registro (ou item de conjunto de resultados representando um registro) dentro do banco de dados Explain, é chamado de " Explain record".

11 - Termination Facility

Termination Facility consiste de um único serviço, Close

O serviço Close permite que o cliente ou servidor terminem abruptamente todas as operações ativas e iniciem o término da Z- Associação.

O serviço Close só pode ser usado quando versão 3 estiver em uso. Nesse caso, seguindo a inicialização, a qualquer hora um pedido Close ou é emitido ou é recebido, tanto o cliente ou o servidor: podem emitir um pedido Close, considere todas as operações ativas serem terminadas abruptamente, espere uma resposta Close (descartando qualquer mensagem interveniente), e considere a Z-Associação fechada; e deve estar preparado para receber um pedido Close, considere todas as operações ativas serem terminadas abruptamente, emita uma resposta Close, e considere a Z-Associação fechada

Operações e Reference-id

Um pedido do cliente por um tipo de operação particular inicia uma operação que é terminada pela respectiva resposta do servidor. Os seguintes tipos de operação são definidos: Init, Search, Present, Delete, Resource-report, Sort, Scan, Extended-services, and Duplicatedetection. Assim cada tipo de pedido de cliente corresponde a um tipo de operação com a exceção dos seguintes tipos de pedido: Trigger-resource-control and Close. Uma operação consiste em um pedido inicial e a resposta de término, junto com qualquer interveniente pedido e resposta de Access-control e Resource-control, pedidos Trigger-resource-control e pedidos Segment. Uma operação é nomeada por um Reference-id pelo cliente, o cliente inclui o Reference-id dentro do pedido inicial, e deve ser incluído dentro de cada mensagem da operação. Se 'operações consecutivas' estão em uso, o parâmetro de Reference-id pode ser omitido no pedido inicial; neste caso o Reference-id é considerado nulo para esta operação, e todas as outras mensagens dessa operação também devem omitir o parâmetro de Reference-id.

Qualquer mensagem enviada do cliente para o servidor ou vice-versa é parte de uma operação (identificada pelo seu Reference-id), com as seguintes exceções: Um pedido ou resposta Close não é parte de qualquer operação.

Se 'operações simultâneas' estão em uso qualquer pedido ou resposta Resource-control or Access-control que não inclui um Reference-id não é parte de uma operação.

Este padrão não assume nenhuma relação entre uma determinada operação e qualquer subsequente operação até mesmo se a operação posterior usa o mesmo

Reference-id. Este padrão não especifica os conteúdos do parâmetro de Reference-id, nem seu significado, exceto à extensão que é usado para se referir a uma operação. Reference-id's sempre são nomeados pelo cliente e só têm significado dentro do sistema de cliente. Desde então nenhuma semântica é atribuída ao Reference-id, ele não tem nenhum tipo de dado incluído e só pode ser descrito como dado binário transparente. Seu tipo ASN.1 é OCTET STRING.

Embora o cliente e servidor devam prover Reference-id como descrito acima, no caso, onde um Reference-id incorreto é fornecido, o receptor do Reference-id incorreto pode ignorar isto simplesmente, ou pode declarar um erro de protocolo (isto é, emitir um Close com justificativa de 'erro de protocolo'). Um exemplo de um Reference-id incorreto seria quando o cliente envia um pedido de Search com um Reference-id e o servidor responde com um Reference-id de um valor diferente, ou nenhum Reference-id. Outro exemplo: o cliente envia um pedido de Search, o servidor envia um pedido de Access-control (com o Reference-id correto) e o cliente envia uma resposta de Access control com Reference-id diferente.

Anexo 2 – Código Fonte da Simulação de Funcionamento do Serviço *Searchlib*

Arquivo: TabelaNomes.java

Descrição: Classe que descreve a estrutura da tabela de nomes de um servidor.

```
public class TabelaNomes {
    private int          nbServ;
    public String        nomeBib[]=new String[12];
    public int          endereco[]=new int[12];
    public int          escopo[]=new int[12];

    public TabelaNomes() {
        nbServ=0;
    }

    public void insereServ(String nm, int ender, int escp) {
        if (nbServ<=9) {
            nomeBib[nbServ]=nm;
            endereco[nbServ]=ender;
            escopo[nbServ]=escp;
            nbServ++;
        } else
            System.out.println("Ultrapassou limite da tabela de nomes");
    }
}
```

Arquivo: SearchLib.java

Descrição: Classe que descreve a estrutura da primitiva de serviço *Searchlib*

```
public class SearchLib {
    public int scope;
    public boolean resultsearch;
    public TabelaNomes resultset;
}
```

Arquivo: DadosCompartilhados.java

Descrição: Classe que implementa a estrutura que será compartilhada pelo Thread Cliente e Thread Servidor.

```
public class DadosCompartilhados{
    SearchLib msg;
    private boolean writeable = true;

    public synchronized void send( SearchLib entrada )
    {
        while ( !writeable ) {
```

```

    try {
        wait();
    }
    catch ( InterruptedException e ) {
        e.printStackTrace();
    }
}
msg = entrada;
writeable = false;
notify();
}

public synchronized SearchLib receive()
{
    while ( writeable ) {
        try {
            wait();
        }
        catch ( InterruptedException e ) {
            e.printStackTrace();
        }
    }
    writeable = true;
    notify();

    return msg;
}
}

```

Arquivo: Cliente.java

Descrição: Classe que implementa o Thread Cliente e a interface com o usuário.

```

import javax.swing.JOptionPane;
public class Cliente extends Thread {
    private DadosCompartilhados dc[];
    private SearchLib searchLib = new SearchLib();

    public Cliente( DadosCompartilhados dadoscomp[] )
    {
        super( "Cliente" );
        dc = dadoscomp;
    }

    public void run()
    {
        String output;
        int serv, escbusc,opc;
        for (;){

```

```

opc=Integer.parseInt(JOptionPane.showInputDialog("Escolha a Opção: \n 1-Efetuar
Busca de Biblioteca 2-Sair"));
if (opc==1){
    serv=Integer.parseInt(JOptionPane.showInputDialog("Escolha o Servidor: \n"+
        "0-Biblioteca Universitária (Servidor de Nomes Nacional)\n"+
        "1-Biblioteca do CTC (Servidor Local) \n"+
        "2-Biblioteca do CCS (Servidor Local) \n"+
        "3-Bib. 1 da Unisul (Servidor de Nomes Local) \n"+
        "4-Bib. 2 da Unisul (Servidor Local) \n"+
        "5-Bib. 1 da UFRGS (Servidor de Nomes Regional) \n"+
        "6-Bib. 2 da UFRGS (Servidor Local) \n"+
        "7-Bib. 1 da UFSM (Servidor de Nomes Local) \n"+
        "8-Bib. 2 da UFSM (Servidor Local) \n"+
        "9-Bib. 1 de BERKELEY (Servidor de Nomes Local) \n"+
        "10-Bib. 2 de BERKELEY (Servidor Local) \n"+
        "11-Biblioteca do Congresso (Servidor de Nomes Global)"));
    escbusc=Integer.parseInt(JOptionPane.showInputDialog("Escolha o Escopo da
Busca: \n 1-Local \n 2-Regional \n 3-Nacional \n 4-Global"));
    searchLib.scope=escbusc;
    do {
        dc[serv].send(searchLib);
        try {
            Thread.sleep( (int) ( Math.random() * 3000 ) );
        }
        catch( InterruptedException e ) {
            System.err.println( e.toString() );
        }
        searchLib = dc[serv].receive();
        output="";
        for (int i=0;i<=11;i++)
            if (searchLib.resultset.nomeBib[i] != null)
                output+=searchLib.resultset.nomeBib[i]+\n";
        if (searchLib.resultsearch==false){ ..
            serv=searchLib.resultset.endereco[0];
            JOptionPane.showMessageDialog(null,"Resultado da
Busca: Negativo - Direcionando Automaticamente para: \n"+output);
        }
    } while (searchLib.resultsearch==false);

    if (searchLib.resultsearch==true)
        JOptionPane.showMessageDialog(null,"Resultado da Busca: Positivo
\n"+output);
}
break;
}
System.exit(0);
}
}

```

Arquivo: Servidor.java

Descrição: Classe que implementa o Thread Servidor.

```
import javax.swing.JOptionPane;
public class Servidor extends Thread {
    private DadosCompartilhados dc;
    private SearchLib searchLib;
    private TabelaNomes tn,tntemp;
    private int servType;
    private String nome;

    public Servidor( DadosCompartilhados dadoscomp, TabelaNomes tabNomes, int st,
String nm)
    {
        super( "Servidor "+nm );
        dc = dadoscomp;
        tn = tabNomes;
        servType = st;
        nome=nm;
    }

    public void run()
    {
        int cont;
        for(;;) {
            searchLib = dc.receive(); ..
            JOptionPane.showMessageDialog(null, Thread.currentThread().getName() +
" recebeu Searchlib");
            searchLib.resultsearch=false;
            if (servType == 0) {
                searchLib.resultset=tn;
            } else {
                cont=0;
                tntemp=new TabelaNomes();
                if (servType==1) {
                    if (searchLib.scope==1){
                        searchLib.resultsearch=true;
                        for (int i=0;i<=9;i++)
                            if (tn.escopo[i]==1){
                                tntemp.nomeBib[cont]=tn.nomeBib[i];
                                cont++;
                            }
                    }
                }
                if (searchLib.scope!=1){
                    searchLib.resultsearch=false;
                    for (int i=0;i<=9;i++)
                        if (tn.escopo[i]==searchLib.scope){
                            tntemp.nomeBib[cont]=tn.nomeBib[i];
                        }
                }
            }
        }
    }
}
```

```

        tntemp.endereco[cont]=tn.endereco[i];
        cont++;
    }
    if (cont==0)
        for (int i=0;i<=9;i++)
            if (tn.escopo[i]==2) {
                tntemp.nomeBib[cont]=tn.nomeBib[i];
                tntemp.endereco[cont]=tn.endereco[i];
                cont++;
            }
    }
    if (servType==2){
        if ((searchLib.scope==1) || (searchLib.scope==2)){
            searchLib.resultsearch=true;
            for (int i=0;i<=9;i++)
                if (tn.escopo[i]==searchLib.scope){
                    tntemp.nomeBib[cont]=tn.nomeBib[i];
                    cont++;
                }
        }
        if ((searchLib.scope!=1) && (searchLib.scope!=2)){
            searchLib.resultsearch=false;
            for (int i=0;i<=9;i++)
                if (tn.escopo[i]==searchLib.scope){
                    tntemp.nomeBib[cont]=tn.nomeBib[i];
                    tntemp.endereco[cont]=tn.endereco[i];
                    cont++;
                }
            if (cont==0)
                for (int i=0;i<=9;i++)
                    if (tn.escopo[i]==3) {
                        tntemp.nomeBib[cont]=tn.nomeBib[i];
                        tntemp.endereco[cont]=tn.endereco[i];
                        cont++;
                    }
        }
    }
    if (servType==3){
        if ((searchLib.scope==1) || (searchLib.scope==2) || (searchLib.scope==3)){
            searchLib.resultsearch=true;
            for (int i=0;i<=9;i++)
                if (tn.escopo[i]==searchLib.scope){
                    tntemp.nomeBib[cont]=tn.nomeBib[i];
                    cont++;
                }
        }
    }
    if ((searchLib.scope!=1) && (searchLib.scope!=2) && (searchLib.scope!=3) ){

```



```

int                                i;
for (i=0;i<=11;i++)
    dc[i] = new DadosCompartilhados();

for (i=0;i<=11;i++)
    tn[i] = new TabelaNomes();

tn[0].inereServ("Biblioteca Universitária - UFSC",0,1);
tn[0].inereServ("Biblioteca do Centro Tecnológico - UFSC",1,1);
tn[0].inereServ("Biblioteca do Centro de Ciências da Saúde - UFSC",2,1);

tn[0].inereServ("Biblioteca Universitária - UFSC",0,2);
tn[0].inereServ("Biblioteca 1 - Unisul",3,2);

tn[0].inereServ("Biblioteca Universitária - UFSC",0,3);
tn[0].inereServ("Biblioteca 1 - UFRGS",5,3);

tn[0].inereServ("Biblioteca do Congresso Americano",11,4);

tn[1].inereServ("Biblioteca Universitária - UFSC",0,1);

tn[2].inereServ("Biblioteca Universitária - UFSC",0,1);

tn[3].inereServ("Bib. 1 - UNISUL",3,1);
tn[3].inereServ("Bib. 2 - UNISUL",4,1);

tn[3].inereServ("Biblioteca Universitária - UFSC",0,2);

tn[4].inereServ("Bib. 1 - UNISUL",3,1);

tn[5].inereServ("Bib. 1 - UFRGS",5,1);
tn[5].inereServ("Bib. 2 - UFRGS",6,1);

tn[5].inereServ("Bib. 1 - UFRGS",5,1);
tn[5].inereServ("Bib. 2 - UFRGS",7,1);

tn[5].inereServ("Biblioteca Universitária - UFSC",0,3);

tn[6].inereServ("Bib. 1 - UFRGS",5,1);

tn[7].inereServ("Bib. 1 - UFSM",7,1);
tn[7].inereServ("Bib. 2 - UFSM",8,1);

tn[7].inereServ("Bib. 1 - UFRGS",5,2);

tn[8].inereServ("Bib. 1 - UFSM",7,1);

tn[9].inereServ("Bib. 1 - BERLELEY",9,1);

```

```

tn[9].insereServ("Bib. 2 - BERKELEY",10,1);

tn[9].insereServ("Biblioteca do Congresso Americano",11,2);

tn[10].insereServ("Bib. 1 - BERLELEY",9,1);

tn[11].insereServ("Biblioteca do Congresso Americano",11,1);

tn[11].insereServ("Bib. 1 - BERKELEY",9,2);
tn[11].insereServ("Biblioteca do Congresso Americano",11,2);

tn[11].insereServ("Biblioteca do Congresso Americano",11,3);

tn[11].insereServ("Biblioteca Universitária - UFSC",0,4);
tn[11].insereServ("Biblioteca do Congresso Americano",11,4);

cliente = new Cliente( dc );

servidor[0] = new Servidor( dc[0], tn[0], 3, "BU-UFSC"); //BU-UFSC
servidor[1] = new Servidor( dc[1], tn[1], 0, "CTC-UFSC"); //CTC-UFSC
servidor[2] = new Servidor( dc[2], tn[2], 0, "CCS-UFSC"); //CCS-UFSC
servidor[3] = new Servidor( dc[3], tn[3], 1, "Bib.1-UNISUL"); //Bib.1-UNISUL
servidor[4] = new Servidor( dc[4], tn[4], 0, "Bib 2-UNISUL"); //Bib 2-UNISUL
servidor[5] = new Servidor( dc[5], tn[5], 2, "Bib 1-UFRGS"); //Bib 1-UFRGS
servidor[6] = new Servidor( dc[6], tn[6], 0, "Bib 2-UFRGS"); //Bib 2-UFRGS
servidor[7] = new Servidor( dc[7], tn[7], 2, "Bib 1-UFSM"); //Bib 1-UFSM
servidor[8] = new Servidor( dc[8], tn[8], 0, "Bib 2-UFSM"); //Bib 2-UFSM
servidor[9] = new Servidor( dc[9], tn[9], 1, "Bib 1-BERKELEY");
servidor[10] = new Servidor( dc[10], tn[10], 0, "Bib 2-BERKELEY");
servidor[11] = new Servidor( dc[11], tn[11], 4, "Bib do Congresso");

for (i=0;i<=11;i++)
    servidor[i].start();

cliente.start();

}
}

```