

Instituto de Matemática e Estatística
Universidade de São Paulo

Sistemas Gerenciadores de Bibliotecas

Dairton Luiz Bassi Filho
Orientador: Eduardo Colli

20 de junho de 2005

Sumário

1	Introdução	3
2	Sistemas Integrados de Bibliotecas	3
2.1	Padrão MARC	4
2.2	Z39.50	5
3	Motivação	5
4	Trabalhos Relacionados	6
4.1	Koha	7
4.1.1	Banco de Dados	8
4.2	LearningAccess ILS	9
4.3	ISIS	10
4.3.1	CDS/ISIS	10
4.3.2	WEBLIS	10
4.4	Gnuteca	11
4.4.1	Banco de Dados	11
4.5	OpenBiblio	12
4.5.1	Banco de Dados	13
4.6	Emilda	14
4.6.1	Banco de dados	15
5	Arquiteturas de Sistemas	15
5.1	Modularização	16
6	Conclusão	16

1 Introdução

A função de uma biblioteca é reunir e organizar livros, revistas, jornais, teses, etc, de forma que seja fácil localizar uma dada informação. Quando o conjunto de itens cresce, os métodos de gerenciamento devem ser revistos adaptados para comportar um volume maior de informações.

O gerenciamento da informação é o que garante que uma biblioteca exerça sua função. Se o gerenciamento não ocorre de forma adequada a biblioteca pode assumir um papel inverso ao que deveria e a localização de uma informação em meio a tantas outras se torna tão difícil e custosa que a biblioteca perde seu papel principal e assume características de um depósito de informações, onde alguns itens são esquecidos e outros simplesmente abandonados.

Além das informações inerentes aos itens bibliográficos, há outro grande conjunto de dados, o das pessoas que interagem com a biblioteca, que são principalmente os usuários e funcionários. O dia-a-dia da biblioteca é constituído de interações entre as pessoas e o conteúdo do acervo, gerando um novo conjunto de informações que também precisa ser administrado.

Para auxiliar no processo de controle das informações um sistema de software faz-se necessário. Idealmente este sistema deve refletir a situação de cada item bibliográfico, saber das operações realizadas por cada usuário e funcionário, auxiliar em tarefas rotineiras e realizar processos automaticamente.

Um software que atua dentro de uma biblioteca auxiliando o gerenciamento das informações é chamado de Sistema Integrado de Biblioteca (*Integrated Library System - ILS*).

2 Sistemas Integrados de Bibliotecas

Um sistema integrado de biblioteca (**ILS**) é um software que oferece suporte aos vários setores da biblioteca. Os serviços essenciais que um sistema deste tipo deve prover são: **Circulação**, **Catálogo** e **OPAC**.

O módulo de *Circulação* é o que compreende as operações realizadas com itens do acervo, gerenciando os empréstimos, devoluções, reservas e renovações. Permitindo identificar onde ou com quem está cada item do acervo, estabelecer e verificar regras para a circulação das obras e atribuir multas e suspensões quando as mesmas não forem cumpridas.

O *Catálogo* objetiva substituir os tradicionais e antigos arquivos de metal com milhares de catálogos de papel usados para localizar um item ou usuário. Em um ILS este serviço deve oferecer diversas formas de pesquisa, permitindo que os bibliotecários e usuários façam consultas de maneira fácil e rápida. O

catálogo também deve permitir que os usuários consultem suas informações e acompanhem a sua situação perante a biblioteca.

O **OPAC** (On-line Public Access Catalog) é a disponibilização do conteúdo dos catálogos em um site na web, de forma que qualquer pessoa, em qualquer lugar e em qualquer horário, possa realizar consultas quanto a existência e disponibilidade dos itens no acervo.

Os campos de busca variam conforme a escopo do sistema, os mais comuns são: ISSN, título, autor, assunto e tipo de material. Permitindo que a busca seja por uma ou mais palavras ou por partes de palavras. Em sistemas mais completos é possível buscar por editora, subassuntos e ano de publicação, além de combinar critérios de busca.

Além destes, os serviços de **Aquisição** e **Weeding** também são desejáveis em ILS's, entretanto esses não estão em um nível de prioridade mais baixo de comparados com Circulação e Catálogo e OPAC, por isso nem todos ILS os provêem.

A *Aquisição* permite à biblioteca planejar seu crescimento, gerenciar os processos de compra, manter registros de doações e também pode, em ILS's mais complexos otimizar e racionalizar a aplicação dos recursos da biblioteca.

O *weeding* [WDG, 2005] é o processo periódico e contínuo de avaliação dos recursos disponíveis na biblioteca com o objetivo de manter o acervo acessível e utilizável. A política de weeding prevê que aos bibliotecários selecionem os livros que serão removidos para dar lugar a novas aquisições.

2.1 Padrão MARC

MARC [MARC, 2005] é a sigla para **MA**chine-**R**eadable **C**ataloging. O MARC surgiu a trinta anos e define um formato de dados para representar os itens da coleção de uma biblioteca. Ele provê mecanismos para que computadores possam usar, trocar e interpretar informação bibliográfica. Seus elementos se tornaram a base dos catálogos usados pelas bibliotecas atualmente.

Library of Congress é o órgão responsável pela definição e manutenção do MARC. Com o passar dos anos o formato do MARC sofreu adaptações, na década de 80 a necessidade de abranger novas informações impulsionou a criação do **USMARC** nos Estados Unidos e no **CAN/MARC** no Canadá. Em 1997 foi criado o **MARC 21** que unificou e acomodou as diferenças entre os formatos americano e canadense.

2.2 Z39.50

O **Z39.50** é um protocolo que permite a busca rápida por obras em servidores de catálogos. As bibliotecas utilizam este protocolo para tornar disponíveis seus catálogos para outras bibliotecas usando o padrão MARC.

3 Motivação

A necessidade de informatizar as coleções está diretamente associada à quantidade de itens dentro dela. A partir do momento que uma pessoa perde tempo para encontrar manualmente um item da coleção faz-se necessário um sistema para gerenciar esta coleção e tornar a localização dos elementos eficaz.

Em bibliotecas as coleções são geralmente grandes e muitas vezes a localização de um item pode exigir algum trabalho e gastar um certo tempo. Isso mostra a necessidade de um ILS com o objetivo de melhorar dois quesitos fundamentais em uma biblioteca, a *produtividade* e a *acessibilidade*.

Os primeiros benefícios que um ILS pode trazer à biblioteca são o aumento da velocidade de realização das tarefas e na eliminação das operações rotineiras realizadas por funcionários.

Muitas tarefas podem ser executadas somente pelo sistema, sem a intervenção de funcionários, reduzindo da quantidade de processos manuais causando o aumento regularidade dos serviços e a diminuição do número de erros humanos.

A acessibilidade ao conteúdo do acervo se reflete de várias formas no modo de trabalho dos bibliotecários aumentando a produtividade. Um mecanismo automático de buscas facilita o acesso a informações, disponibiliza novos conjuntos de dados, aumenta o volume de dados analisados e amplia o controle sobre o acervo. O acesso à informação torna-se mais fácil, mais amplo e mais profundo. Os serviços da biblioteca tornam-se precisos e o modo de executá-los mais eficiente, eliminando a duplicação de trabalho. Além disso, a acessibilidade amplia seus limites quando o mecanismo de busca compartilha dados e pesquisa em outras bibliotecas.

Atualmente existem vários tipos de ILS comerciais que exigem um alto investimento para a sua aquisição. Este investimento não está ao alcance de pequenas bibliotecas, acervos escolares, igrejas e outras instituições públicas. Ademais, seria difícil para estas instituições sustentar os gastos de manutenção e investir em atualizações das novas versões do ILS adquirido. Esta dificuldade motiva esta pesquisa por sistemas de código aberto, sobretudo, outra forte razão para a realização deste trabalho é a necessidade de ILS's

que dêem suporte às características de bibliotecas universitárias, pois estas possuem características particulares que estão associadas, principalmente, à grande variedade de tipos de usuários deste ambiente. É necessário considerar alunos de graduação, mestrado, doutorado e especiais, professores do mesmo instituto da biblioteca, de outros institutos e visitantes, e também é preciso tratar empréstimos entre as bibliotecas das faculdades dentro da universidade, o que faz da biblioteca um novo tipo de usuário.

Outra peculiaridade de bibliotecas universitárias é a vasta coleção de *periódicos* que compreende revistas científicas, jornais e títulos que temporariamente publicam novos volumes e fascículos, que são de suma importância para os pesquisadores e alunos da universidade.

A coleção de *periódicos* tem um comportamento diferente porque cada periódico é formado por subitens, os *volumes*, e estes são compostos por *fascículos*. Além disso, temporariamente são adquiridos novos fascículos e volumes, isto torna o periódico um item em constante expansão o que significa que seu comportamento é completamente diferente dos itens estáticos do acervo. Essas características tornam os periódicos uma coleção de itens que requer que o ILS ofereça novos nas funcionalidades para tratá-los adequadamente.

O gerenciamento da coleção de periódicos e o conhecimento sobre quais fascículos e volumes estão na coleção da biblioteca são tarefas importantes e necessitam do auxílio de um software para agilizar sua execução. Portanto, um dos pontos de pesquisa será por sistemas que também dêem suporte ao gerenciamento de coleções de periódicos.

4 Trabalhos Relacionados

Nos últimos cinco anos algumas iniciativas em diferentes partes do mundo começaram a produção de sistemas para gerenciamento de bibliotecas. Nesta seção faremos uma breve explanação sobre as implementações mais relevantes que já possuem um conjunto razoável de funcionalidades desejadas. Atualmente vários desses projetos já possuem ou almejam a produção dos módulos de *aquisição*, *catálogo*, *circulação* e *OPAC*. A análise deste trabalho será focada principalmente em três quesitos: as funcionalidades oferecidas, as tecnologias empregadas e a modelagem dos dados. Em particular, uma atenção especial será dada à modelagem da base de dados, valorizando otimizações, flexibilidade e extensibilidade.

4.1 Koha

Koha ¹ é um software neozelandês desenvolvido pela *Katipo Communications* ² para a *Honowhenua Library Trust* ³. Sua primeira versão ficou pronta em 2000 e atualmente está na versão 2.2, disponível no *SourceForge* sob licença GNU General Public License (GPL).

O Koha ⁴ é uma aplicação feita em *Perl*, roda dentro de um servidor de aplicações e usa um banco de dados relacional para persistir os dados. Desta forma ele depende dos seguintes softwares livres:

- Apache Web Server
- MySQL ou outro Gerenciador de banco de dados relacional baseado em SQL.
- Interpretador Perl com os seguintes módulos:
 - Date::Manip
 - DBI
 - Set::Scalar
 - DBD::mysql (ou outro correspondente ao banco de dados usado)
 - AuthenDBI - para autenticação no banco de dados

Os recursos oferecidos pelo Koha compreendem os principais módulos desejados em um ILS. Ele possui um catálogo on-line, além dos módulos de circulação e aquisição [Eyler, 2003].

As principais características funcionais são:

- interface web simples para consultas de usuários e funcionários.
- controle de circulação
- controle de aquisição
- controle simplificado de aquisição para pequenas bibliotecas
- catalogação de web sites, tratando-os como itens do acervo
- interface para administração

¹<http://www.kohadocs.org>

²<http://www.katipo.co.nz>

³<http://www.library.org.nz/cgi-bin/koha/opac-main.pl>

⁴<http://www.koha.org>

- suporte ao formato MARC
- pesquisa através do protocolo Z39.50
- gerador de código de barras

O software visa atender bibliotecas de pequeno a grande porte, assim, é possível optar ou não pelo uso do formato MARC. A forma simplificada, sem o padrão MARC, é ideal para pequenas instituições interessadas apenas na catalogação e gerenciamento do acervo. Quando o formato MARC é usado, torna-se possível importar e exportar arquivos de catálogos.

O gerenciamento de usuários permite a classificação em categorias a fim de atribuir permissões diferentes. Os funcionários podem ser divididos em grupos conforme as funções exercidas dentro da biblioteca e assim garantir o acesso apenas aos recursos desejados.

A equipe da biblioteca pode ter acesso a interface administrativa via web sem que usuários externos tenham acesso. Isto é feito separando a interface administrativa em outra aplicação no servidor e disponibilizando-a em uma porta do servidor aberta apenas para intranet.

Através da interface administrativa o bibliotecário pode acessar a ficha de cada usuário e visualizar seus empréstimos e reservas correntes.

A camada de software que atua sobre o banco de dados oferece muitas funcionalidades desejadas para o gerenciamento de bibliotecas, de forma que o sistema oferece ferramentas para os setores de aquisição, catalogação e circulação.

4.1.1 Banco de Dados

O script criador do banco de dados, é usado em MySQL mas não utiliza restrições de integridade entre as entidades. Isto torna o banco de dados praticamente um conjunto de tabelas independentes.

A modelagem do acervo trata da mesma forma todos os tipos de itens bibliográficos e por conseqüência os armazena juntos nas mesmas entidades. Este tratamento uniforme não considera particularidades de alguns tipos de itens. Se novos dados precisarem ser considerados para algum tipo de item bibliográfico será preciso criar novos campos nas entidades que representam o acervo. Porém, esta operação incluirá estes campos em todos os tipos de itens. Assim, todos os itens, independente de usá-los ou não, alocarão espaço para os novos atributos, havendo desperdício de espaço em muitos casos.

Além da falta de restrições de integridade há sérios problemas de normalização de tabelas, muitas entidades possuem campos do tipo *string* para armazenar dados que poderiam ser representados por outras entidades. Esta

modelagem provoca replicação de dados pois as mesmas *strings* são armazenadas em vários registros e torna as operações de busca altamente ineficientes se comparadas com buscas sobre campos de tipos numéricos [Date, 2000], [Teorey, 1999].

A replicação de *strings* aponta para outro problema, um modelo de dados limitado que não representa pequenas entidades. Esta limitação tira do banco de dados a responsabilidade de fornecer valores corretos para os campos do tipo *string*. Esta tarefa fica portanto para o sistema de software que atua sobre a base dados. Algumas vezes a interface do usuário contém valores *hard coded* que são enviados para a persistência no banco. Quando a interface não contém esses valores, geralmente, fica a cargo do usuário digitá-los, neste caso a chance de digitações incorretas, abreviações ou falta de acentuação são altíssimas, provocando diferenças no conteúdo de campos que deveriam ser iguais. As conseqüências são buscas sem sucesso, dificuldade para encontrar os registros e o diminuição da acurácia em estatísticas baseadas nos dados do banco.

A representação das pessoas também se reduz a uma única entidade que contém todos os tipos de usuários. Esta entidade contém muitos atributos e a maioria dos registros deixa muitos campos vazios. Nela acontece o mesmo problema do acervo, tratamento uniforme para todos os tipos de pessoas gerando uma entidade com muitos atributos, que desperdiça espaço e ao longo do tempo a tendência é o surgimento de problemas de performance.

4.2 LearningAccess ILS

O **LearningAccess** ⁵ foi financiado por doações e pelo governo do estado de Washington. Ele é também conhecido como **Koha West** e **OpenBook**. Este projeto é uma ramificação do *Koha* que migrou a codificação dos scripts Perl pra PHP e implementou já no seu início suporte ao padrão MARC 21. Apesar da separação, vários módulos do Koha escritos em Perl que acessam o banco de dados foram mantidos.

O escopo inicial era pequenas bibliotecas e acervos escolares, entretanto, a arquitetura atual é capaz de suportar mais de 250.000 itens bibliográficos. O sistema é composto por três módulos: o módulo de usuários (OPAC), um módulo de catálogos e o módulo de circulação. Todos com interface web e com suporte multi-lingue, inicialmente disponível em inglês, espanhol e francês.

O módulo Catálogo possui interface para buscas no OCLC ⁶ com o pro-

⁵<http://www.learningaccess.org/website/techdev/ils.php>

⁶<http://www.oclc.org>

protocolo Z39.50. Dados podem ser importados no formato MARC 21 porém a exportação deve ser implementada na versão 1.0.

O módulo circulação está em fase de desenvolvimento. Quando completo irá oferecer serviços de empréstimo/devolução através de código de barras e permitir aos usuários fazer reservas dos itens desejados. Também faz parte dos planos implementar o módulo aquisição mas ainda não há definições ou detalhes a respeito dele.

Apesar dos bons planos de implementação, o projeto está parado sem novas versões a mais de dois anos. A versão atual possui muitos bugs e ainda não é recomendada para o uso em ambiente de produção.

4.3 ISIS

O **WEBLIS** é uma aplicação da família *ISIS*⁷, desenvolvida pela UNESCO, que conta com vários softwares de gerenciamento de informações, em particular de dados bibliotecários.

4.3.1 CDS/ISIS

O **CDS/ISIS** (*Computerized Documentation Service/Integrated Set of Information System*) é um software desenvolvido pela UNESCO desde 1985. A primeira versão é o **MICROISIS** que roda sobre o DOS e possui interface texto, posteriormente foi lançado o **WINISIS** com interface gráfica para Windows. Sua principal função é prover o armazenamento e recuperação de informações visando atender às necessidades de instituições públicas, escolas, igrejas e bibliotecas, principalmente de países em desenvolvimento.

Atualmente a UNESCO está avaliando a possibilidade de tornar o CDS/ISIS um software de código aberto, isso o inclui nesta análise pois pelo largo tempo de uso esta seria a implementação mais confiável.

4.3.2 WEBLIS

O **WEBLIS** é um ILS freeware baseado no CDS/ISIS e implementado sobre a plataforma WWW-ISIS que é um software desenvolvido para atuar como servidor web com bases de dados ISIS.

Este software tem sua expansão limitada pela indisponibilidade do código fonte e pelo vínculo com a base de dados CDS/ISIS. Atualmente o WEBLIS está disponível apenas em inglês e oferece os módulos de Catalogação, OPAC, Empréstimos e Estatísticas.

⁷<http://www.unesco.org/isis>

O módulo de catalogação permite o armazenamento de livros, séries, jornais, artigos, vídeos, fotos, áudio e slides. O OPAC permite buscas simples e avançadas no catálogo e em enciclopédias on-line. O módulo de circulação controla empréstimos, renovações, reservas e faz cobrança automática, enviando e-mail ou gerando cartas de cobrança.

O módulo de estatísticas armazena os dados calculados em outra base de dados ISIS, paralela à base que contém os dados principais.

4.4 Gnuteca

O **Gnuteca**⁸ é um software brasileiro desenvolvido pelo Centro Universitário Univates⁹ usando o ambiente CDS/ISIS, o que torna fácil a migração de dados neste formato. O software é destinado ao controle de circulação dos itens da biblioteca e está sob licença CC-GNU-GPL. Atualmente o desenvolvimento do Gnuteca é feito através de uma parceria entre Univates e a Solis¹⁰, uma empresa formada por alunos, professores e ex-funcionários da Univates.

O Gnuteca foi desenvolvido em PHP, apoiado pelo framework Miolo¹¹ que usa apenas conceitos de orientação a objetos em PHP. A arquitetura do sistema é cliente/servidor. Na máquina onde funciona o servidor é necessário um web server Apache e um banco de dados, o padrão é o PostgreSQL mas pode ser usado outro SGBD que permita armazenar dados sem tamanho pré-definido e ofereça suporte a subcampos. A aplicação instalada nas máquinas cliente também usa PHP com Miolo e o PHP-GTK¹² para a execução da interface gráfica.

As funcionalidades do Gnuteca são focadas nos módulos de catálogo e circulação, englobando controle de empréstimos, reservas, renovações e histórico de empréstimos, e oferecendo a opção de importar e exportar catálogos no formato MARC 21.

4.4.1 Banco de Dados

Os scripts de criação do banco de dados quase não usam importantes recursos que os sistemas gerenciadores de bancos de dados relacionais (SGBD) oferecem. O relacionamento entre as tabelas, na maioria dos casos, não impõe restrições de integridade, isso não assegura a consistência dos dados abrindo

⁸<http://www.gnuteca.org.br>

⁹<http://www.univates.br>

¹⁰<http://www.solis.coop.br>

¹¹<http://www.miolo.org.br>

¹²<http://gtk.php.net>

brechas para a persistência de dados inválidos. Em contrapartida, dois recursos sofisticados são usados: os *subcampos* para representar os dados do padrão MARC; e *SEQUENCE's* em algumas tabelas para controlam a geração de *ids*. Esses recursos não são implementados por todos gerenciadores de bancos de dados, reduzindo a quantidade de SGBD compatíveis.

O detalhamento dos elementos é bem pequeno, muitos atributos não são considerados, assim diversos tipos de itens do acervo não possuem todos seus dados. A modelagem trata entidades importantes do sistema de forma monolítica. As *pessoas* e o *conteúdo do acervo* são representados por poucas entidades, isso faz com que todo registro possua algum campo que nunca é utilizado. Esta modelagem reflete objetos grandes em memória que sempre ocupam mais espaço do que deveriam.

A parte do banco que representa as pessoas é basicamente representada por uma única tabela. Esta modelagem obriga a existência dos campos de todos os tipos de pessoas físicas e jurídicas para todos os registros. Isso tende a trazer problemas de flexibilidade e extensibilidade [Elmasri and Navathe, 1989], a expansão do sistema abrangendo novos dados aumentará cada vez mais o desperdício de espaço e tornará cada vez mais difícil a remodelagem pois todo o resto do sistema foi construído para funcionar com um modelo muito simples.

4.5 OpenBiblio

O **OpenBiblio** ¹³ é um software desenvolvido desde de 2002 e está disponível no Sourceforge sobre licença GPL. Seu objetivo é gerenciar acervos bibliotecários, a atual versão é recomendada para pequenos acervos, oferece os módulos de circulação, OPAC e catalogação, permitindo a importação de dados no padrão USMARC.

O OpenBiblio é uma aplicação web, que roda no Apache, escrita inteiramente em PHP que usa o MySql como gerenciador banco de dados.

Algumas de suas características:

- permissões administrativas por módulo
- geração de etiquetas para os itens da coleção
- geração de cartas para os usuários
- relatórios sobre o acervo, os usuários e os empréstimos
- interface amigável e fácil de usar

¹³<http://obiblio.sourceforge.net>

As funcionalidades são limitadas e ainda há vários bugs espalhados pelo sistema, entretanto é uma solução que já apresenta um bom conjunto de recursos que serão úteis a pequenas bibliotecas.

Algumas funções são limitadas ou estão ausentes. As principais deficiências são:

- No módulo OPAC, não é possível a busca por ISSN, os parâmetros de busca não são flexíveis, somente é permitido buscas por título, autor e assunto.
- O módulo circulação só permite busca por usuários, e mesmo assim somente pelo *número de usuário* ou pelo sobrenome. Seria importante, neste módulo as buscas por obra e exemplar.
- Não é possível fazer reservas dos itens.

4.5.1 Banco de Dados

O banco de dados usado por padrão é o MySQL, assim como o *Koha*, os scripts de geração da base de dados do *OpenBiblio* não utilizam restrições de integridade entre as entidades. A decisão de fazer scripts simples para a geração do banco de dados permite que praticamente qualquer outro SGBD seja capaz de executá-los sem gerar problemas de incompatibilidade.

O modelo de dados possui um baixo nível de detalhamento, todos os tipos de itens bibliográficos são tratados da mesma forma, sem que muitos dados sejam considerados. As entidades, em geral, não possuem muitos atributos, que é bom do ponto de vista de desempenho, mas considerando o escopo do sistema, o modelo de dados representa pobremente o mundo real e não pode ser considerado representativo para bibliotecas grandes e médias.

A modelagem simplória provoca limitações no modelo dados e tende a trazer complicações quando o sistema evoluir. No futuro, a consideração de novos tipos de elementos obrigará os desenvolvedores a escolher entre: *a)* manter a modelagem atual e provocar o inchaço das entidades, ou, *b)* fazer uma remodelagem provocando de grandes mudanças estruturais.

Se durante a expansão do modelo de dados for mantida a postura de armazenar vários tipos nas mesmas entidades, então, estas sofrerão aumento no seu número de atributos e passarão a desperdiçar espaço, a exemplo da modelagem de *Koha* e do *Gnuteca*. Caso o crescimento seja acompanhado de remodelagens, a base de dados terá um desempenho melhor mas será preciso mais trabalho para adaptar a camada de negócios ao novo modelo de dados [Sommerville, 2001].

4.6 Emilda

O Emilda ¹⁴ é desenvolvido pela empresa finlandesa Realnode ¹⁵ desde 2003, em menos de 2 anos se apresenta como uma boa solução para gerenciamento de circulação, catálogos e usuários. Ele é um sistema com interface web, e usa MySQL versão 3.x ou superior. Sua codificação é, na maior parte, em PHP e Perl, todavia, também há código em C++ e em JavaScript.

As principais funcionalidades estão associadas aos módulos de circulação, gerenciamento de usuários e catalogação. O sistema usa os formatos internacionais MARC, Z39.50 e **LDAP** para auxiliar na catalogação. O LDAP (*Lightweight Directory Access Protocol*) é um protocolo para busca e atualização de dados em diretórios que roda sobre TCP/IP. Este protocolo usa os diretórios X.500 mas se comporta de maneira muito mais simples que o protocolo original, o *Directory Access Protocol* (DAP).

Muitos recursos interessantes estão associados ao gerenciamento de usuários e permitem que eles interajam com a biblioteca, os que mais se destacam são:

- *Permissões* - há três tipos de acesso, administradores, funcionários e usuário da biblioteca. Cada categoria consegue apenas realizar as operações compatíveis com seu papel.
- *Grupos* - os administradores podem reunir os usuários em grupos para facilitar o gerenciamento.
- *Mensagens* - os usuários podem trocar mensagens dentro do sistema e fazer solicitações aos administradores, assim como os administradores podem enviar avisos e informativos aos usuários.
- *Avaliação* - os usuários podem associar comentários e notas aos itens da biblioteca, de forma que os demais usuários possam consultá-las.
- *Busca externa* - através dos protocolos Z39.50 e LDAP podem ser feitas buscas nas grandes bibliotecas do mundo de forma muito eficiente.
- *Solicitações* - os usuários podem solicitar a aquisição de novas obras. Para facilitar e evitar pedidos com dados insuficientes da obra, é possível fazer a solicitação consultando catálogos obtidos através das redes Z39.50 e LDAP

¹⁴<http://www.emilda.org>

¹⁵<http://www.realnode.fi>

4.6.1 Banco de dados

A exemplo dos outros ILS não há imposição de integridade referencial entre as entidades. E em alguns casos *strings* são armazenadas em campos repetidamente, quando uma melhor solução seria criar uma entidade com estas *strings* e fazer referência a elas através de ids.

Outra fraqueza é a falta do controle de transações. Sem este controle, um erro no banco de dados ou na conexão com o servidor pode tornar os dados inconsistentes. Somando a falta de integridade referencial e a falta de transações, a base de dados fica exposta a eventuais erros de programação, pois o BD não tem nenhuma proteção contra operações errôneas.

A modelagem do acervo é abrangente e consegue representar os campos MARC e LDAP, em particular, a estrutura usada para armazenar os campos MARC coloca *nome do campo* e *valor* como valores dos atributos de uma entidade, esta modelagem é genérica e flexível o suficiente para comportar quaisquer eventuais mudanças. O lado negativo é a perda de performance durante as buscas. [Ramakrishnan, 1997].

5 Arquiteturas de Sistemas

A análise dos ILS's mostrou uma forte preferência por aplicações web, está foi a escolha do *OpenBiblio*, *Emilda*, *LearningAccess*, *WEBLIS* e *Koha*. Esta opção está associada à necessidade do oferecimento da interface web para o módulo OPAC. Assim, para manter a interface unificada a tendência é fazer todo o sistema no formato de aplicação web. Esta opção de design é prática pois torna o ILS acessível a partir de qualquer máquina na internet. Entretanto a decisão de fazer uma aplicação web deve vir acompanhada de um nível maior de segurança, da mesma forma que o ambiente web torna a aplicação facilmente acessível para os usuários, ela também fica exposta a usos mal intencionados.

A arquitetura com aplicações *cliente/servidor* do Gnuteca vai de encontro ao modelo de aplicação web. Ela é mais adequada aos módulos usados apenas pelos funcionários da biblioteca pois tais módulos só devem ser acessados de dentro da biblioteca. ele abre mão o OPAC e ganha um grau de segurança maior [Berson, 1996].

A arquitetura adotada pelo *Koha* oferece a melhor solução pois pondera acessibilidade e segurança. A divisão do sistema em dois módulos atende aos requisitos da biblioteca sem abrir mão da segurança. O módulo com o OPAC que fica disponível na web, o outro com as funções administrativas, que fica disponível apenas na intranet.

5.1 Modularização

A modularidade de um sistema é representada pela independência funcional dos seus componentes [Pressman, 1992]. O desenvolvimento de aplicações modularizadas tornou-se um aspecto decisivo no projeto de sistemas. De forma que esta característica influi diretamente na qualidade do software final.

A facilidade de expansão e manutenção também está associada à modularização. A independência entre os módulos possibilita a modificação ou a exclusão de um módulo sem afetar os demais.

O estudo das necessidades sistêmicas de uma biblioteca mostra a clara divisão das funcionalidades e independência entre as tarefas. Esta característica motiva que um ILS seja constituído de módulos que possam operar de forma independente. Os grupos de operações que até aqui chamados *módulos*: circulação, aquisição, catálogo, etc são módulos que devem ser considerados na arquitetura do sistema e não apenas uma serem tratados como conjuntos de operações da interface. [Barrera et al., 2004], [Ferreira and Finger, 2000] e [Ferreira and Busichia, 1999].

6 Conclusão

Através da pesquisa por softwares capazes de gerenciar bibliotecas encontramos boas iniciativas onde os módulos de catalogação, OPAC e circulação estão mais maduros ao ponto de serem usados em ambientes de produção, no entanto, ainda é comum a ausência de algumas funcionalidades e nenhum deles se caracteriza como uma solução completa, nos projetos menos evoluídos a ausência dos módulos e a extrema simplicidade torna-os bem aplicáveis apenas em contextos restritos.

Na tabela abaixo podemos visualizar de forma comparativa os recursos que cada ILS oferece.

	Koha	LAccess	Gnuteca	WEBLIS	OBiblio	Emilda
Catálogo	X	X	X	X	X	X
OPAC	X	X		X	X	X
Circulação	X	X*	X	X	X	X
Aquisição	X					
Periódicos						
MARC	X	X	X		X	X
Universit.						

*incompleto

Nenhum dos ILS avaliados compreende a gama de tipos de usuários encontrados no ambiente universitário. Assim como, nenhum deles trata da hierarquia dos *itens periódicos com volumes e fascículos*. A forma como os sistemas tratam os itens periódicos é limitada a um simples rótulo, sem proporcionar as funcionalidades desejadas para esse tipo de item.

O principal motivo dos ILS não atenderem a todos os quesitos desejados é o modelo de dados. Todos os softwares analisados apresentam modelagens simplistas que não usam boas estruturas de dados para tornar o modelo genérico e abrangente.

A maioria dos ILS usa o MySQL, este SGBD suporta *foreign keys* desde a versão 3.23.44, de novembro de 2001, entretanto os ILS não utilizam este recurso ganhando a portabilidade do modelo de dados para um número maior de SGBDs. No entanto, considerando a estabilidade do MySQL e o suporte a *foreign keys* pela maioria dos SGBD, uso deste recurso é, além de desejável, fortemente justificável, visando tornar a base de dados livre de inconsistências.

Referências

- [Barrera et al., 2004] Barrera, J., Junior, R. M. C., Ferreira, J. E., and Gubitoso, M. D. (2004). An environment for knowledge discovery in biology. *Journal of Computers in Biology and Medicine*, pages 371–459.
- [Berson, 1996] Berson, A. (1996). *Client/Server Architecture*. McGraw-Hill, New York.
- [Date, 2000] Date, C. J. (2000). *An Introduction to Database Systems*. Addison-Wesley, Reading, 5 edition.
- [Elmasri and Navathe, 1989] Elmasri, R. and Navathe, S. (1989). *Fundamentals of Database Systems*. Benjamins/Cummings, Redwood City.
- [Eyler, 2003] Eyler, P. (2003). Koha: a gift to libraries from new zeland. *Linux J.*, 2003(106):1.
- [Ferreira and Busichia, 1999] Ferreira, J. E. and Busichia, G. (1999). Database modularization design for the construction of flexible information systems. *Proceedings IEEE for the IDEAS99*, pages 415–422.
- [Ferreira and Finger, 2000] Ferreira, J. E. and Finger, M. (2000). Controle de concorrência e distribuição de dados: Teoria clássica, suas limitações e extensões modernas. *XII Escola de Computação*, pages 73–83.

- [MARC, 2005] MARC (2005). Marc standards. <http://www.loc.gov/marc>.
Último acesso em: 12/06/2005.
- [Pressman, 1992] Pressman, R. S. (1992). *Software Engineering: a Practitioner's Approach*. McGraw-Hill, Inc.
- [Ramakrishnan, 1997] Ramakrishnan, R. (1997). *Database Management Systems*. McGraw-Hill.
- [Sommerville, 2001] Sommerville, I. (2001). *Software Engineering*. Addison-Wesley, 6th edition.
- [Teorey, 1999] Teorey, T. (1999). *Database Modeling and Design: the E-R Approach*. Morgan Kaufman.
- [WDG, 2005] WDG (2005). Weeding policy. <http://www.dlapr.lib.az.us/cdt/weeding.htm>. Último acesso em: 12/06/2005.