

Mário Roberto Ferreira

PROJETO BIBLIOS
PROPOSTA DE BANCO DE DADOS

Monografia apresentado ao Departamento de
Ciência da Computação da Universidade Federal
de Lavras como parte das exigências do curso de
Pós-Graduação *Lato Sensu* Administração em
Redes Linux, para obtenção do título de
especialista.

Orientador

Prof. Msc Joaquim Quinteiro Uchôa

LAVRAS
MINAS GERAIS - BRASIL
2004

Mário Roberto Ferreira

PROJETO BIBLIOS
PROPOSTA DE BANCO DE DADOS

Monografia apresentado ao Departamento de
Ciência da Computação da Universidade Federal
de Lavras como parte das exigências do curso de
Pós-Graduação *Lato Sensu* Administração em
Redes Linux, para obtenção do título de
especialista.

Aprovada em 25 de abril de 2004

Prof. Msc Ricardo Martins de Abreu Silva

Prof. Samuel Pereira Dias

Prof. Msc Joaquim Quinteiro Uchôa
(Orientador)

LAVRAS
MINAS GERAIS - BRASIL

Dedico este trabalho
primeiramente
ao Senhor Jesus,
por transformar
coisas impossíveis em possíveis
e à minha amada esposa.

Resumo

Os sistemas gerenciadores de bancos de dados, SGBD, têm-se estabelecido como tecnologia de armazenamento de dados. É uma tecnologia robusta que garante segurança, consistência e integridade dos dados. A parte disso surgiu a necessidade de implementação de um aplicativo para gestão de bibliotecas aproveitando todo o potencial dos SGBD's. E assim substituir o aplicativo de catalogação bibliográfica, CDS/ISIS, de tecnologia limitada em base de dados textuais.

O Projeto Biblios é um projeto de *software* para catalogação bibliográfica e empréstimo de acervo voltado para pequenas e médias bibliotecas. O aplicativo será disponibilizado gratuitamente sob a bandeira do *Software Livre*.

Sumário

1 – INTRODUÇÃO.....	9
2 - REVISÃO DA LITERATURA.....	13
2.1 – CDS/ISIS	13
2.1.1 - <i>Características do CDS-ISIS</i>	13
2.1.2 - <i>Acesso Multiusuário</i>	14
2.1.3 - <i>Cópia de Segurança e Recuperação</i>	15
2.2 – OUTRAS OPÇÕES DE SOFTWARE DE GESTÃO BIBLIOGRÁFICA	16
2.3 – SISTEMA GERENCIADOR DE BANCO DE DADOS – SGBD	17
2.3.1 - <i>Limitações do SGBD PostgreSQL</i>	21
3 - METODOLOGIA APLICADA.....	23
4 – PROJETO BÍBLIOS	27
4.1 – PROJETO CONCEITUAL	30
4.2 – DEFINIÇÃO DA BASE DE DADOS.....	31
4.2.1 - <i>Catálogo bibliográfico</i>	32
4.2.2 - <i>Acervo</i>	39
4.2.3 - <i>Empréstimo</i>	41
4.3 - ACESSO MULTIUSUÁRIO.....	45
4.4 - CÓPIA DE SEGURANÇA E RECUPERAÇÃO	46
4.5 – POLÍTICAS DE ACESSO	46
5 – CONCLUSÃO	49
6 - REFERÊNCIAS BIBLIOGRÁFICAS	51
ANEXO A - ESQUEMA BANCO DE DADOS BIBLIOS	54
ANEXO B - LISTA DE ABREVIATURAS	63

Lista de Figuras

Figura 1: Camadas de acesso ao banco de dados	18
Figura 2: QBE pgaccess	28
Figura 3: Resultado da consulta feita pelo QBE.....	29
Figura 4: Caso de Uso do Projeto Biblios	31
Figura 5: Módulos do projeto Biblios	32
Figura 6: ER do módulo de catalogação bibliográfica	33
Figura 7: ER do módulo acervo	39
Figura 8: ER do módulo de empréstimo	41
Figura 9: Tela principal do Projeto Biblios.....	50

Lista de Tabelas

Tabela 1: Restrições e limitações do sistema CDS/ISIS	13
Tabela 2: Limitações do PostgreSQL.....	22
Tabela 3: Tabela categoria	33
Tabela 4: Índices da tabela categoria	34
Tabela 5: Tabela editora	34
Tabela 6: Índices da tabela editora	34
Tabela 7: Tabela idioma	35
Tabela 8: Índices tabela idioma	35
Tabela 9: Tabela tipobiblio	35
Tabela 10: Índices da tabela tipobiblio	35
Tabela 11: Tabela autor.....	36
Tabela 12: Índices da tabela de autor.....	36
Tabela 13: Tabela assunto	36
Tabela 14: Índices da tabela assunto.....	36
Tabela 15: Tabela listabiblio	36
Tabela 16: índices da tabela listabiblio	37
Tabela 17: Tabela listabiblioassunto.....	38
Tabela 18: Tabela listabiblioautor	38
Tabela 19: Regras da tabela listabiblioautor.....	38
Tabela 20: Tabela listabibliopalavra	38
Tabela 21: Índices da tabela listabibliopalavra	38
Tabela 22: Tabela biblioteca.....	39
Tabela 23: Índices da tabela biblioteca.....	39
Tabela 24: Tabela acervo	40
Tabela 25: Índices da tabela acervo.....	40
Tabela 26: Regras da tabela acervo	40
Tabela 27: Tabela tipousuario	42
Tabela 28: Índices da tabela tipousuario	42
Tabela 29: Regras da tabela tipousuario	42
Tabela 30: Tabela curso	42
Tabela 31: Índices da tabela curso.....	42
Tabela 32: Tabela usuario	43
Tabela 33: Indices da tabela usuario	44
Tabela 34: Regras da tabela usuario	44
Tabela 35: Tabela calendario	44
Tabela 36: Tabela emprestimo	44
Tabela 37: Índices da tabela emprestimo	45
Tabela 38: Regras da tabela emprestimo	45
Tabela 39: Permissão de acesso.....	47

1 – INTRODUÇÃO

Nos últimos anos, tem-se observado o crescimento do sistema operacional Linux como uma alternativa aos sistemas proprietários. O Linux é um sistema robusto e com baixo custo de aquisição e manutenção. Com esse crescimento, por sua vez, alavancou uma demanda por vários tipos de *software* como: processadores de textos, gráficos, sistemas de contas a pagar e receber, entre outros. Nessa situação encontram-se os *software* de gestão de bibliotecas, visto que o *software* mais utilizado pelas bibliotecas, [LIMA (1999)], o CDS/ISIS, utiliza justamente os sistemas operacionais proprietários, como o WindowsTM ¹.

Por outro lado, atualmente existem para o GNU/Linux compiladores e interpretadores de alta qualidade e confiabilidade para o desenvolvimento de programas, podendo ser citados o PHP², GCC³, Perl⁴, entre outros. Aliado a essas ferramentas de desenvolvimento, destacam-se os sistemas gerenciadores de bancos de dados (SGBD's), como por exemplo, o PostgreSQL⁵, o MySQL⁶ e o Firebird⁷. Atualmente existem bancos ocupando área em disco nas casas dos Terabytes⁸ [PostgreSQL (2003)].

Também existe todo um contexto e liberdade necessários para a criação de um *software* de gestão de bibliotecas para o ambiente Linux. Liberdade por

¹ WindowsTM Sistema Operacional produzido pela Microsoft

² PHP <http://www.php.net>

³ GCC <http://www.gnu.org/software/gcc/gcc.html>

⁴ Perl <http://www.perl.org>

⁵ PostgreSQL <http://www.postgresql.org>

⁶ MySQL <http://www.mysql.com>

⁷ Firebird <http://firebird.sourceforge.net/>

⁸ Terabyte equivale a 1.024 Gigabytes ou 1.48.576 Megabytes

existir disponíveis para utilização *software* de desenvolvimento e SGBD gratuitos e passíveis de serem alterados, com código fonte aberto e livre para alteração e cópia. Para maiores detalhes sobre as liberdades apresentadas pelo *Software Livre* veja [Uchôa (2003)].

Esses programas são disponibilizados geralmente através de um tipo especial de licença, como o GNU/GPL [GNU (2003)] ou o BSD [BSD (2003)]. Estas licenças têm em seus termos: os direitos, garantias e principalmente a declaração de *Software Livre*. Esta declaração define a possibilidade do *software* ser utilizado livremente e mesmo adequá-lo às necessidades locais.

Nesse contexto o Projeto Biblios surgiu como uma proposta de banco de dados para o programa de gestão de bibliotecas. O projeto originou-se no Departamento de Ciência da Computação da Universidade Federal de Lavras/MG com o objetivo de ser uma opção viável de *software* em substituição ao sistema CDS/ISIS. O Projeto Biblios tem os seguintes fatores motivadores:

- utilizar SGBD, bancos de dados relacionais, em comparação ao sistema CDS/ISIS que é um banco de formato texto;
- utilizar arquitetura multiplataforma, ou seja, a possibilidade de ser utilizado em vários sistemas operacionais, comerciais ou não;
- ter mecanismos de segurança, aumentando a qualidade da informação armazenada;
- reduzir a manutenibilidade, ou seja, reduzir o trabalho em correções futuras ocasionadas por falhas ou novas adequações. Princípios de manutenibilidade e qualidade de *software* encontram-se estabelecidas em [ISO 9126 (1991)].

Para a realização deste trabalho foram expostas no capítulo 2, as limitações e os problemas de segurança do sistema CDS/ISIS em comparação com os SGBD's. Nos capítulos 3 e 4 serão propostos uma metodologia de trabalho e uma base de dados implementada no SGBD PostgreSQL como estrutura base para o desenvolvimento do aplicativo.

2 - REVISÃO DA LITERATURA

2.1 – CDS/ISIS

O sistema CDS/ISIS é o *software* mais utilizado para catalogação bibliográfica [LIMA (1999)] devido a sua interface intuitiva e pela flexibilidade na criação de base de dados. O ISIS é distribuído de forma gratuita pela UNESCO através da BIREME [BIREME (2003)]. Caso a biblioteca implemente algum programa de pesquisa bibliográfica, o ISIS disponibiliza uma biblioteca denominada ISIS_DLL, para acessar a base de dados.

2.1.1 - Características do CDS-ISIS

Os usuários podem criar novos campos e armazená-los em formato texto, sem controle de tamanho mínimo ou máximo de quantidade de caracteres, letras ou números. A Tabela 1 refere-se às limitações do sistema CDS/ISIS, [ISIS (2003)], veja que a base de dados apesar de ser textual limita o tamanho do registro em 8000 caracteres e o total da base em 500 MB.

Tabela 1: Restrições e limitações do sistema CDS/ISIS

Número máximo de bases de dados	Ilimitado
Máximo de registros na base de dados, dentro do limite de 500 MB	16 milhões
Tamanho máximo do registro	8000 caracteres
Máximo de campos (definíveis na FDT), excluindo repetições de campos repetitivos	200 campos
Número máximo de linhas da FST	300 linhas
Tamanho máximo do campo	8000 caracteres
Máximo de campos em uma página de planilha	19 campos
Máximo de páginas em uma planilha	20 páginas
Tamanho máximo de um formato de exibição	6000 caracteres
Máximo de palavras proibidas	799 palavras

Fonte: IBICT-ISIS/2004

O ISIS possui uma interface simples, apresentando recursos como: consulta booleana, exportação dos dados formatados nos padrões ISO e XML.

Atualmente o ISIS, está disponível principalmente para o sistema operacional WindowsTM. Ele também apresenta uma versão para o sistema operacional UnixTM e tem recursos para interface Web, porém é uma versão paga e dependente de recursos proprietários, como por exemplo, a linguagem de desenvolvimento ASP pertencente à Microsoft.

O sistema CDS/ISIS faz o gerenciamento e armazenamento dos dados através da definição de alguns arquivos, os quais são relacionados pelo programa para manutenção e visualização do registro. Isso limita a manipulação dos dados por outros aplicativos pois a única forma de acessá-los é através do aplicativo CDS/ISIS.

cds.mst	Arquivo principal, <i>masterfile</i> , responsável pela associação dos campos e das informações armazenadas.
cds.fdt	Tabela de definição de campos
cds.pft	Arquivo com formato padrão de impressão
cds.ifp	Arquivo de pesquisa invertida
.fmt	Tabela de <i>layouts</i> de folhas de dados
.fst	Tabela de seleção de campos

Arquivos de imagens e sons são apenas referenciados pelo sistema, e não são armazenados diretamente na base de dados.

2.1.2 - Acesso Multiusuário

O diretório, onde encontram-se os arquivos executáveis (aplicativos) e a base de dados, deve ser compartilhado no servidor para que outros usuários

possam montá-los (mapeá-los) em seus computadores.

A segurança fica comprometida, pois os usuários que compartilham o sistema terão acesso amplo às estruturas. Comandos de manipulação de arquivos como exclusão podem comprometer todo o sistema.

Um outro problema é a proliferação de vírus de computador pela rede, exigindo antivírus atualizado a todos os usuários, o que nem sempre é possível. O problema está no atraso no lançamento de uma vacina eficaz para o vírus, deixando que o vírus se prolifere e provoque maiores danos.

2.1.3 - Cópia de Segurança e Recuperação

Os arquivos, aplicativos e base de dados, deverão ser freqüentemente copiadas através de aplicativos do próprio sistema operacional, exemplo xcopy, copy, zip, entre outros. Além disso, deve-se tomar o cuidado para não deixar de copiar o arquivo invertido [ISIS (2003)].

Outra recomendação refere-se em realizar a cópia sem que o programa esteja funcionando, ou seja, *backup off-line*. Isto é interessante para não haver perda de sincronia entre a cópia e a base de dados, além de evitar corrupções. A perda de sincronia ocorre quando existem arquivos abertos de forma exclusiva pelo programa CDS/ISIS e esses são impedidos de serem copiados pelo programa de cópia.

Observe que nesse caso existe um problema para o administrador do sistema, pois terá que solicitar que os usuários saiam do sistema para realizar a cópia.

A recuperação será necessária quando ocorrer alguma corrupção da base de dados. Segundo o [Oráculo (2004)], as principais ocorrências de corrupções das bases de dados do ISIS referem-se a:

- perda dos arquivos mestre ou invertido;
- inserção de caracteres especiais na base de dados, caracteres abaixo do código ASCII 32;
- exclusão da base de dados.

A recuperação exige uma *backup* confiável e consistente, caso contrário poderão ocorrer perdas irrecuperáveis da base de dados.

2.2 – Outras opções de *software* de gestão bibliográfica

Além do CDS/ISIS, existem outras soluções para o controle bibliográfico: opções comerciais, opções *in-house* e também código aberto. As opções comerciais são inviáveis pelo seu alto custo. As opções *in-house*, aquelas produzidas localmente por algumas bibliotecas, não são disponibilizadas, sendo restritos ao uso interno, ver, por exemplo [UFOP (2004)], [UFPE (2004)].

A opção código aberto atualmente disponível denomina-se Gnuteca [Gnuteca (2004)]. O Gnuteca utiliza o SGBD PostgreSQL porém não utiliza o que há de melhor na tecnologia de banco de dados. O Gnuteca em primeira análise apresenta alguns problemas estruturais em suas bases como:

- ausência de chave primária e índices únicos – possibilita a duplicação de valores em campo identificado no sistema como “chave”;
- ausência de integridade referencial – possibilita a geração de registros órfãos em tabelas vinculadas por relacionamento;
- campos com preenchimento incorreto, provocado por falta de validações.

O Gnuteca, como o CDS/ISIS, traz para o aplicativo a responsabilidade de manter a consistência dos dados, normalmente realizada pelo SGBD. Caso o aplicativo possua alguma falha poderá provocar corrupção da base de dados pela falta de consistência administrada pelo SGBD.

2.3 – Sistema Gerenciador de Banco de Dados – SGBD

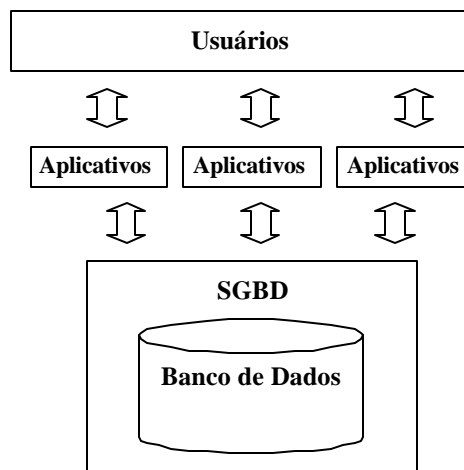
O SGBD é o *software* responsável pela interface entre o banco de dados e a aplicação, como mostrado na Figura 1. O SGBD é uma coleção de programas que controla a criação, consulta e manutenção dos dados armazenados no banco de dados. Observe que os aplicativos somente têm acesso ao banco de dados através do SGBD via comandos SQL (*Structured Query Language*).

As vantagens na utilização do SGBD sobre outros formatos de bancos de dados são:

- facilitação do desenvolvimento de sistemas – a parte de manipulação, armazenamento e compartilhamento dos dados é delegada ao SGBD;

- transparência dos dados com relação aos aplicativos – os aplicativos não se preocupam como os dados são armazenados, pois interagem com o SGBD via comandos SQL;
- maior integridade dos dados – os dados armazenados devem atender ao objetivo dos campos evitando absurdos, por exemplo: sexo com preenchimento diferente de F ou M;
- maior controle sobre a segurança dos dados armazenados – são atribuídos aos usuários permissões de acesso com o objetivo de impedir pessoas não autorizadas consultar ou mesmo alterar os dados.

Figura 1: Camadas de acesso ao banco de dados



A definição de uma base de dados para um SGBD é vista como uma coleção de tabelas que estabelecem relacionamento entre si. Cada tabela possui um ou mais campos e pelo menos um campo identificador, chave primária, que identifique o registro de forma única.

As relações entre as tabelas são realizadas através de alguns campos identificados como chaves primárias e chaves estrangeiras. As relações são regidas por uma determinada cardinalidade e definem quantos registros de uma tabela se relacionam com outras tabelas. As possibilidades de relações entre as tabelas são basicamente três: um para um; um para muitos e muitos para muitos [Manzanos (2003)].

Para eliminar redundâncias utiliza-se um processo de normalização, criando novas tabelas e relacionamentos. Normalmente isso acontece, quando existem relacionamentos muito para muitos, justificando uma tabela intermediária entre as entidades.

Em campos com entradas específicas pode-se criar regras, como por exemplo: único - *unique* – somente poderá haver um registro com o valor informado; requerido – *not null* – campo com preenchimento obrigatório ou mesmo criando regras específicas através dos comandos SQL *create rule* ou *check* [PostgreSQL (2003)].

A organização da base de dados dá-se através de três modelos: modelo físico, modelo lógico e modelo de visões.

O modelo físico é a forma como o SGBD armazena os arquivos no disco e como se interage com o sistema operacional. O modelo lógico é a organização das bases de dados em tabelas e suas relações. É através desse modelo que o administrador interage com o SGBD, manipula as estruturas e altera os dados armazenados.

O modelo de visões corresponde a criação de *views*, onde o usuário não tem acesso detalhado ao modelo lógico, mas de partes dos dados formatados para seu uso. Este recurso tem como aplicação prática o impedimento de execuções de comandos prejudiciais ao sistema, visto que não são fornecidas informações detalhadas das estruturas existentes.

Na definição de uma base de dados trabalha-se o modelo lógico, definindo as tabelas e suas relações, através de comandos SQL [Ramalho (1999)]. O SQL é uma linguagem poderosa de consulta, associada ao SGBD. Além de ser utilizada para a criação das tabelas e seus relacionamentos, também é utilizada para a manipulação dos dados, ou seja, manutenção e consultas.

Date [Date (2000)] define algumas características dos SGBD's que destacam sobre outros programas de banco de dados. Observe que estas características são simplificadas pela sigla **ACID**:

- **atomicidade** (*atomicity*) - algumas operações sobre os dados devem ser processadas de forma conjunta e indivisível. Isto é, ou todas as operações são realizadas ou nenhuma é realizada;
- **consistência** (*consistency*) – como consequência da atomicidade, a execução de uma transação ocorre de forma isolada, promovendo a consistência dos dados;
- **isolamento** (*isolation*) – controla o acesso aos dados por diversos usuários, sem perder a consistência dos dados. Cada transação não toma conhecimento de outras transações concorrentes do sistema;

- **durabilidade** (*durability*) – promove a persistência e a durabilidade após as alterações dos dados em um meio de armazenamento confiável, podendo se recuperar depois de algum tipo de falha física ou queda de energia.

Como consequência dessas características os SGBD's relacionais consolidaram-se sobre os bancos de dados tradicionais principalmente no desenvolvimento de aplicativos. Muitos dos controles de manipulação de dados, manutenção e consultas foram delegados ao SGBD, facilitando e reduzindo o trabalho dos desenvolvedores.

2.3.1 - Limitações do SGBD PostgreSQL

Com relação às limitações do PostgreSQL, comparando com o CDS/ISIS, o registro poderá chegar a 1.6 TB além de não existir limites de quantidade, conforme indicado na Tabela 2.

Arquivos de imagens e sons são armazenados diretamente no banco de dados através de campos com tipo OID. O tipo OID é semelhante ao tipo BLOB (*Binary Long Object*) de outros bancos.

Tabela 2: Limitações do PostgreSQL

Tamanho máximo do banco de dados	Ilimitado (há bancos com 4 TB)
Tamanho máximo de uma tabela	16 TB
Tamanho máximo de um registro	1.6 TB
Tamanho máximo de um campo	1 GB
Quantidade máxima de registros	Ilimitado
Quantidade máxima de campos em uma tabela	250 - 1600 dependendo do tipo da coluna
Quantidade máxima de índices em uma tabela	Ilimitado

Fonte: Site oficial do banco PostgreSQL [Postgresql (2003)]

3 - METODOLOGIA APLICADA

Os programas necessários para o desenvolvimento e implantação foram escolhidos baseados nos seguintes critérios: confiabilidade, qualidade, código aberto, gratuito e multiplataforma. Com essas condições o Projeto Biblios tem a portabilidade de ser utilizado em diversos sistemas operacionais, principalmente GNU/Linux e WindowsTM.

O Projeto Biblios é uma aplicação do tipo cliente servidor, ou seja, existem três camadas de trabalho para a sua implementação: banco de dados, servidor Web e browser do usuário. Os programas necessários para a implementação dessas camadas foram assim escolhidos:

Servidor

- **servidor SGBD PostgreSQLTM**, banco de dados relacional robusto, está de acordo com a característica ACID, padrão SQL ANSI⁹ 89/92/98, escrito em C/C++, sem custos de licenciamento e de atualizações, disponibiliza utilitários gráficos para o gerenciamento das bases de dados e ampla documentação através de manuais e tutoriais;
- **servidor Web Apache**, [Apache (2003)], existem vários outros servidores de HTTP, porém a escolha do Apache deveu-se pelas seguintes razões: desempenho, facilidade na instalação /

⁹ ANSI – *American National Standard Institute*

configuração, quantidade de sistemas operacionais suportados e suporte a criptografia de páginas.

- **linguagem de desenvolvimento PHP**, [PHP (2003)], totalmente integrado e de fácil configuração com o servidor Web Apache. Tem bibliotecas nativas de acesso ao SGBD PostgreSQL.

Cliente

- **navegador ou *browser***, é o aplicativo necessário para acessar as páginas disponibilizadas na Web pelo Projeto Biblios. Cita-se alguns navegadores como o Mozilla¹⁰ ou o Netscape¹¹, ambos gratuitos e disponíveis para vários sistemas operacionais.

Gerador de modelos relacionais

- **UML**, *Unified Modeling Language*, na geração de diagramas de caso de uso, utilizou-se um gerador de UML [UML (2004)], realizado pelo aplicativo DIA [DIA (2003)]. A opção pelo padrão de modelagem UML deve-se pela facilidade e flexibilidade na montagem dos diagramas.

O diagrama de casos de uso descreve e define os requisitos funcionais do aplicativo. São definidos atores que iniciam e interagem com o sistema. Os casos de uso definem uma ação ou comportamento do sistema.

¹⁰ Mozilla <http://www.mozilla.org>

¹¹ Netscape <http://www.netscape.com>

- ER, modelo entidade relacionamento, demonstra todo o esquema do modelo lógico de um banco de dados e seus relacionamentos. O ER utiliza retângulos para representar as entidades, losangos para representar os relacionamentos e linhas para ligar as entidades aos relacionamentos.

4 – PROJETO BÍBLIOS

O principal objetivo do Projeto Biblios é utilizar todos os recursos disponibilizados pelo SGBD com o propósito de obter maior segurança e consistência das bases de dados. Essa necessidade é vital para a definição do aplicativo pois os controles oferecidos pelo CDS/ISIS são limitados e não oferecem os mesmo recursos.

Para implementar as bases de dados no SGBD considerou-se os principais pontos críticos comparados aos bancos de dados textuais, citados por Manzanos [Manzanos (2003)], que são :

- facilidade em pesquisar a base de dados;
- armazenar o conteúdo de um campo com tamanho variável;
- utilizar subcampos.

Para realizar a pesquisa o SGBD utiliza a linguagem SQL, através do comando *select* e o CDS/ISIS utiliza comandos embutidos no programa. Veja que apesar de existir uma certa complexidade nos comandos SQL, isso pode ser resolvido com a utilização de uma interface de pesquisa às bases de dados.

Uma opção alternativa para os usuários é utilizar o programa auxiliar que acompanha o PostgreSQL, no caso o *pgaccess* que disponibiliza uma interface gráfica para consultas, identificada como QBE (*query by example*), mostrado nas Figura 2 e Figura 3.

Construtor visual de consultas

Criar tabela: Mostra SQL Executar SQL Salvar no construtor de consultas

editora

- id_editora
- nomecompleto
- nomeabreviado
- endereco
- complemento
- cep
- cidade
- uf
- pais
- telefone
- fax
- site
- email
- observacao

listabiblio

- id_listabiblio
- titulo
- subtitulo
- descpatr
- isbn-issn
- is_categoria
- is_editora
- cutter
- cdd
- volume
- numero
- edicao
- capitulos
- qtdepag
- tipo
- is_idioma
- localpublicacao
- abstract
- anopublicacao
- conteudo
- tempo

Campo	titulo	isbn-issn	nomecompleto	nomeabreviado
Tabela	listabiblio	listabiblio	editora	editora
Ordena	unsorted	unsorted	unsorted	unsorted
Critério				= 'Atlas'
Resultado	Sim	Sim	Sim	Sim

Figura 2: QBE pgaccess

titulo	isbn-issn	nomecompleto	nomeabreviado
Desvendando o Linux	121212121	Editora Atlas Ltda	Atlas

Figura 3: Resultado da consulta feita pelo QBE

Com relação em armazenar campos com conteúdo variável o SGBD PostgreSQL apresenta dois tipos de campos, *varchar* e *text*, os quais têm o limite máximo de armazenamento em 1 GB, o que vai muito além dos restritos 8000 bytes para o registro do CDS/ISIS. O tipo de dados, *varchar* ou *text*, não restringe a criação de índices para o campo, otimizando o tempo de pesquisa.

A utilização de subcampos será resolvida com o processo de normalização. Como exemplo cita-se a tabela *listabiblio* que terá a tabela relacionada *listabiblioautor*. A tabela-filho terá o limite de 2^2 registros para cada registro da tabela principal, o que significa que cada bibliografia poderá ter 2^{32} autores.

Foram criados relacionamentos com integridade referencial para impedir a ocorrência de registros órfãos. Este vínculo é uma relação realizada pela chave

estrangeira, *foreign key*, na tabela-filho com a chave primária, *primary key*, na tabela-pai.

Campos que possuem entradas específicas foram atribuídas validações através do comando SQL, *check*. Como exemplo cita-se o campo *tipo* da tabela *listabiblioautor* limitadas pelas entradas: “1” - autor principal, “2” - colaborador e “3” - outras participações na obra.

Campos com entrada obrigatória foram criados com a opção *not null*, ou seja o campo deverá ser preenchido com algum valor.

4.1 – Projeto conceitual

O projeto conceitual do Projeto Biblios refere-se aos diagramas entidade relacionamento (ER) e casos de uso. Esses diagramas disponibilizam uma visão geral do sistema sem tratar de forma detalhada as estruturas e as funcionalidade do banco de dados.

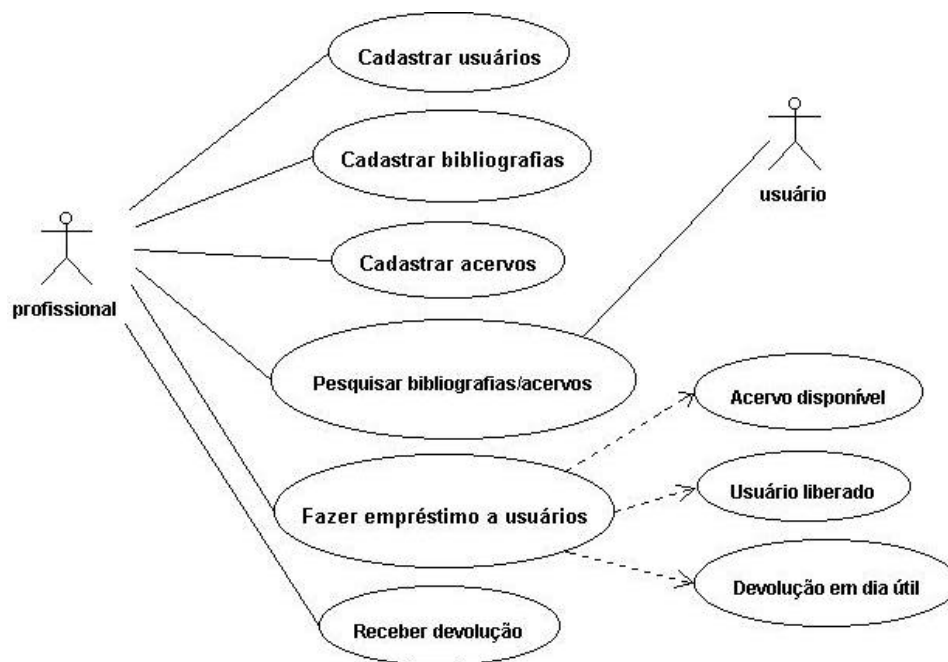


Figura 4: Caso de Uso do Projeto Biblios

4.2 – Definição da base de dados

A definição do dicionário das bases de dados, do Projeto Biblios foi estruturada baseado em três módulos, como mostra a Figura 5.

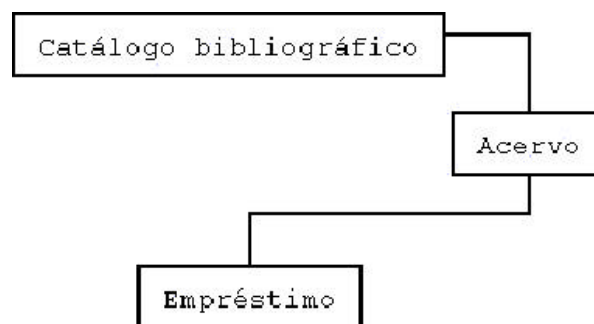


Figura 5: Módulos do projeto Biblios

4.2.1 - *Catálogo bibliográfico*

O catálogo bibliográfico é responsável em armazenar os dados referentes à bibliografia, como mostra a Figura 6. Nele podem ser lançadas informações de livros, revistas, periódicos, cd's , fitas de vídeo, conforme especificado na tabela *tipobiblio*. A tabela *listabiblio* é a principal tabela deste módulo.

A catalogação bibliográfica ocorrerá de forma independente da catalogação patrimonial realizada pelo módulo acervo. A vantagem desse processo é a realização de pesquisas em variadas bibliografias mesmo que não exista o exemplar no acervo da biblioteca.

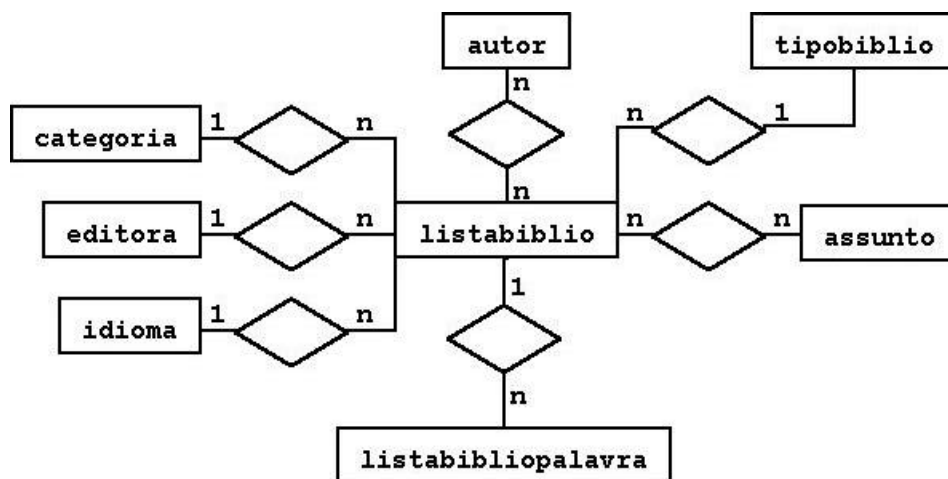


Figura 6: ER do módulo de catalogação bibliográfica

As tabelas deste módulo e a identificação de seus campos, são assim descritos, observando que os campos identificados como chave primária, *primary key*, e com código seqüencial definem os códigos que serão criados automaticamente pelo sistema através de autonumeração.

- **Categoria** – Identifica uma categoria à bibliografia, um outro sinônimo a esta estrutura poderia ser tipos ou classes. Exemplo de categorias: Física, Estudos Sociais, Psicologia, etc. O objetivo desta estrutura é agrupar bibliografias por áreas distintas, como mostra a Tabela 3.

Tabela 3: Tabela categoria

	Campo	Tipo	Tam**	Req.***	Único****	Descrição
PK*	id_categoria	Número-int4	4	Sim	Sim	Código seqüencial
	categoria	Varchar	100	Sim	Sim	Descrição da categoria

* Pk – Primary Key - identifica a chave primária

** Tam – Tamanho do campo, espaço ocupado fisicamente em bytes.

*** Req – Identifica os campos com preenchimento obrigatório

**** Único – Identifica campos que não poderão existir repetições de seus conteúdos em outros registros.

Tabela 4: Índices da tabela categoria

Nome	Campos	Classificação
idxu_categoria	categoria	Asc*

* Ascendente

- **Editora** – Identifica as editoras que produzem o material bibliográfico, livros, revistas, periódicos, etc. Observe que há a opção de armazenar o nome completo e um nome abreviado, ou seja, um nome fantasia que facilita a identificação da editora, como mostra a Tabela 5.

Tabela 5: Tabela editora

	Campo	Tipo	Tam	Req	Único	Descrição
Pk	id_editora	Número-int4	4	Sim	Sim	Código seqüencial
	nomecompleto	Varchar	100	Sim	Sim	Nome completo da editora
	nomeabreviado	Varchar	100	Sim	Sim	Nome fantasia
	endereco	Varchar	50			Endereço
	complemento	Varchar	40			Complemento do endereço
	cep	Varchar	9			CEP
	cidade	Varchar	40			Cidade
	uf	Char	2			UF
	pais	Varchar	40			Pais
	telefone	Varchar	15			Telefone
	fax	Varchar	15			Fax
	site	Varchar	100			Site Internet
	email	Varchar	60			Email
	observação	Text	n*			Dados complementares

* n – sem limites de tamanho

Tabela 6: Índices da tabela editora

Nome	Campos	Classificação
idxu_nomecompleto	nomecompleto	Asc
idxu_nomeabreviado	nomeabreviado	Asc

- **Idioma** – Identifica o idioma. Exemplo Alemão, Inglês, Francês, etc, como mostra a Tabela 7.

Tabela 7: Tabela idioma

	Campo	Tipo	Tam	Req	Único	Descrição
Pk	id_idioma	Número-int4	4	Sim	Sim	Código seqüencial
	idioma	Varchar	100	Sim	Sim	Idioma

Tabela 8: Índices tabela idioma

Nome	Campos	Classificação
idxu_idioma	idioma	Asc

- **Tipobiblio** – Identifica o tipo da bibliografia. Exemplo: Livro, revista, escultura, cd, vhs, etc, como mostra a Tabela 9

Tabela 9: Tabela tipobiblio

	Campo	Tipo	Tam	Req	Único	Descrição
Pk	id_tipobiblio	Número-int2	4	Sim	Sim	Código seqüencial
	tipobiblio	Varchar	100	Sim	Sim	Descrição do tipo da bibliografia

Tabela 10: Índices da tabela tipobiblio

Nome	Campos	Classificação
idxu_tipobiblio	tipobiblio	Asc

- **Autor** - Tabela de autores – Para facilitar a identificação de autores, criou-se nesta tabela o campo apelido, armazenando o nome mais conhecido do autor, como mostra a Tabela 11.

Tabela 11: Tabela autor

	Campo	Tipo	Tam	Req	Único	Descrição
Pk	id_autor	Número-int4	4	Sim	Sim	Código seqüencial
	autor	Varchar	100	Sim	Sim	Nome do autor
	apelido	Varchar	50	Sim		Apelido do autor

Tabela 12: Índices da tabela de autor

Nome	Campos	Classificação
idx_apelido	apelido	Asc
idxu_autor	autor	Asc

- **Assunto** – Tabela de assuntos, associa à bibliografia os assuntos que ela aborda. Exemplo: Linux, sistema operacional, etc, como mostra a Tabela 13.

Tabela 13: Tabela assunto

	Campo	Tipo	Tam	Req	Único	Descrição
Pk	id_assunto	Número-int4	4	Sim	Sim	Código seqüencial
	assunto	Varchar	100	Sim	Sim	Descrição do assunto

Tabela 14: Índices da tabela assunto

Nome	Campos	Classificação
idxu_assunto	assunto	Asc

- **Listabiblio** – Principal tabela deste módulo, responsável pela catalogação das bibliografias, como mostra a Tabela 15

Tabela 15: Tabela listabiblio

	campo	tipo	tam	req	único	descrição
Pk	id_listabiblio	Número-int4	4	Sim	Sim	Código seqüencial
	título	Varchar	n	Sim		Título da obra
	subtítulo	Varchar	n			Subtítulo
	isbn_issn	Varchar	30			Identificação do livro (isbn)

						ou revista (issn)
Fk*	is_categoria	Número-int4	4	Sim		FK(categoria) – Categoria do acervo
Fk*	is_editora	Número-int4	4			FK(editora) – Editora do livro/revista/periódico
	cutter	Varchar	30			Cutter
	cdd	Varchar	30			CDD – Classificação de autores
	cdu	Varchar	30			CDU – Classificação de autores
	volume	Varchar	30			Identificação do volume
	numero	Varchar	20			Ordem do volume
	edicao	Varchar	20			Edição
	capitulos	Número-int4	4			Qtde de capítulos
	qtdepag	Número-int4	2			Qtde de Páginas
	is_tipobiblio	Número-int2	1	Sim		Tipo da bibliografia
Fk*	is_idioma	Número-int4	4	Sim		FK(idioma) – Idioma
	localpublicacao	Varchar	70			Local onde foi publicado o artigo
	abstract	Text	n			Sinopse do acervo
	anopublicacao	Número-int4	2	Sim		Ano da publicação
	conteudo	Text	n			Dados complementares, poderão ser lançados neste campo, o sumário, um resumo, etc.
	tempo	tempo	8			Tempo de duração

* Fk – Foreign Key – Chave estrangeira

Tabela 16: índices da tabela listabiblio

Nome	Campos	Classificação
idxu_titulo	titulo	Asc
idx_subtitulo	subtitulo	Asc
idx_cdd	cdd	Asc
idx_cdu	cdu	Asc
idx_isbn_issn	isbn_issn	Asc
idx_cutter	cutter	Asc

- **Listabiblioassunto** – Atribui o assunto à bibliografia, como mostra a Tabela 17

Tabela 17: Tabela listabiblioassunto

	campo	tipo	tam	req	único	descrição
PK*	is_listabiblio	Número-int4	4	Sim	Sim	FK(listabiblio) - Código da bibliografia
PK	is_assunto	Número-int4	4	Sim	Sim	FK(assunto) - Código do assunto

* Chave primária composta por dois campos

- **Listabiblioautor** – Atribui os autores da bibliografia identificando o tipo de participação na obra, como mostra a Tabela 18

Tabela 18: Tabela listabiblioautor

	campo	tipo	tam	req	único	descrição
PK	is_listabiblio	Número-int4	4	Sim	Sim	FK(listabiblio) - Código da bibliografia
PK	is_autor	Número-int4	4	Sim	Sim	FK(autor) – Código do autor
	tipoautor	Número-int2	2	Sim		Tipo do autor: 1-Autor principal 2 – colaborador 3- outros

Tabela 19: Regras da tabela listabiblioautor

Nome	Campos	Regra
reg_tipoautor	tipoautor	>= 1 e <= 3

- **Listabibliopalavra** – Identifica as palavras chaves que identificam a obra. Exemplo. configuração servidor linux, dns, gateway, rede, etc, como mostra a Tabela 20

Tabela 20: Tabela listabibliopalavra

	campo	tipo	tam	req	único	descrição
PK	id_palavra	Número-int4	4	Sim	Sim	Código sequencial
	is_listabiblio	Número-int4	4	Sim	Sim	FK(listabiblio) - Código da
	palavra	Varchar	100	Sim	Sim	palavra chave

Tabela 21: Índices da tabela listabibliopalavra

Nome	Campos	Classificação
idxu_listabiblio_palavra	palavra	Asc
	is_listabiblio	Asc

4.2.2 - Acervo

Esse módulo é responsável por identificar o patrimônio bibliográfico de cada biblioteca. Nele é atribuído uma identificação patrimonial, tomo, aos acervos existentes, como mostrado na Figura 7.

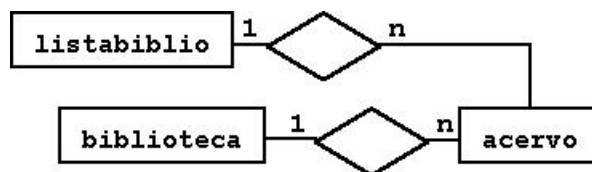


Figura 7: ER do módulo acervo

- **Biblioteca** – Identifica as bibliotecas físicas existentes. Os acervos são atribuídos as bibliotecas, como mostra a. Tabela 22.

Tabela 22: Tabela biblioteca

	campo	tipo	tam	req	único	descrição
Pk	id_biblioteca	Número-int4	4	Sim	Sim	Código Sequencial
	biblioteca	Varchar	100	Sim	Sim	Nome da biblioteca
	telefone1	Varchar	14			Telefone 1
	telefone2	Varchar	14			Telefone 2

Tabela 23: Índices da tabela biblioteca

Nome	Campos	Classificação
idxu_biblioteca	biblioteca	Asc

- **Acervo** – Esta tabela identifica fisicamente o bem patrimonial, associando o catálogo bibliográfico ao patrimônio de uma biblioteca, como mostra a Tabela 24.

Tabela 24: Tabela acervo

	campo	tipo	tam	requerido	único	descrição
Pk	id_acervo	Número-int4	4	Sim	Sim	Código sequencial
Fk	is_listabiblio	Número-int4	4	Sim		FK(listabiblio) - Código da bibliografia
	tombo	Varchar	15			Código patrimonial
	dttombo	Data	8			Data do tombamento
	tipoaquisicao	Número-int2	1	Sim		1-Aquisição 2-Doação 3-Empréstimo
	dtbaixa	Data	8			Data da baixa
	tipobaixa	Número-int2	1			1-Estragado 2-Desaparecido 3-Devolução do Empréstimo
	status	char	1	Sim		1-Liberado 2-Emprestado 3-Bloqueado 4-Baixado 5-Manutenção 6-Emprestado a outra biblioteca
	dados	Varchar	70			Dados adicionais do acervo
	codigo	Varchar	20			Identificação acervo
Fk	is_biblioteca	Número-int4	4			FK(Biblioteca) - Biblioteca proprietária do acervo

Tabela 25: Índices da tabela acervo

Nome	Campos	Classificação
idx_tombo	tombo	Asc

Tabela 26: Regras da tabela acervo

Nome	Campos	Regra
reg_tipoaquisicao	tipoaquisicao	>= 1 e <= 3
reg_tipobaixa	tipobaixa	>=1 e <= 3
reg_status	status	>=1 e <= 6

4.2.3 - Empréstimo

Esse módulo é responsável pelo controle de empréstimo de acervo a usuários da biblioteca, Figura 8.

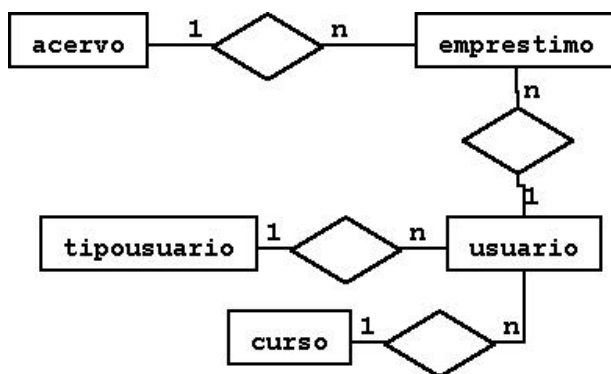


Figura 8: ER do módulo de empréstimo

A política de empréstimo será controlada pela tabela *tipousuario*, através de dois mecanismos: quantidade de dias para ficar com o acervo emprestado e limite na quantidade de acervos emprestados num mesmo período.

Associado ao módulo de empréstimo, haverá o controle de feriados através da tabela *calendário*, necessária para controlar a data de devolução do acervo, impedindo a devolução em domingos e feriados.

- **Tipousuario** – Classifica os usuários por tipo e define a política de empréstimo, quantidade de dias de empréstimo e quantidade máxima de acervos a serem emprestados, como mostra a Tabela 27.

Tabela 27: Tabela tipousuario

	campo	tipo	tam	req	único	descrição
Pk	id_tipousuario	Número-int4	4	Sim	Sim	Código sequencial
	tipo	Varchar	50	Sim	Sim	Tipo do usuário
	qtddias	Número-int2	2	Sim		Dias empréstimo
	qtdeacervo	Número-int2	2	Sim		Máximo de acervo a ser emprestado

Tabela 28: Índices da tabela tipousuario

Nome	Campos	Classificação
idxu_tipousuario	tipo	Asc

Tabela 29: Regras da tabela tipousuario

Nome	Campos	Regra
reg_qtddias	qtddias	> 0
reg_qtdeacervo	qtdeacervo	> 0

- **Curso** – Identifica o curso em que o usuário está atualmente matriculado. O objetivo desta tabela é agrupar os usuários por áreas de formação.

Tabela 30: Tabela curso

	campo	tipo	tam	req	único	descrição
Pk	id_curso	Número-int4	4	Sim	Sim	Código sequencial
	curso	Varchar	50	Sim	Sim	Descrição do curso

Tabela 31: Índices da tabela curso

Nome	Campos	Classificação
idxu_curso	tipo	Asc

- **Usuário** – Identifica as pessoas, usuários, que poderão solicitar empréstimos na biblioteca, com dados pessoais, nome, endereço, documentos e também o

curso que atualmente esteja matriculado. O usuário terá um login de acesso para realizar consultas à base de dados, como mostra a Tabela 32.

Tabela 32: Tabela usuario

	campo	tipo	tam	req	único	descrição
Pk	id_usuario	Número-int4	4	Sim	Sim	Código seqüencial
	is_tipousuario	Número-int4	4	Sim		FK(tipousuario) - Tipo do usuário
	nome	Varchar	100	Sim		Nome do usuário
	endereco	Varchar	100			Endereço
	complemento	Varchar	40			Complemento do endereço
	bairro	Varchar	50			Bairro
	cidade	Varchar	40			Cidade
	uf	Char	2			UF
	cep	Varchar	8			Cep
	telefone	Varchar	14			Telefone
	celular	Varchar	14			Celular
	dtnasc	Data	8			Data de nascimento
	sexo	char	1			Sexo do usuário
	email	Varchar	60			Email
	is_curso	Número-int4	4			FK(curso) – Curso de graduação quando aluno
	matricula	Varchar	30			Código da matrícula
	cpf	Varchar	11			Cpf
	rg	Varchar	20			Documento de Identidade
	orgexped	Varchar	15			Órgão Expedidor RG
	bloqueado	Boolean	1			Usuário está bloqueado
	dtbloq	Data	8			Data do bloqueio
	motivobloq	Varchar	60			Motivo do bloqueamento
	login	Varchar	15	Sim	Sim	Login de conexão ao sistema
	senha	Varchar	60			Senha
	dtinclusao	Data	8			Data da inclusão no cadastro
	is_biblioteca	Número-int4	4			FK(biblioteca) Biblioteca que está lotado se tipousuario = funcionário
	ac_biblioteca	boolean	1	Sim		Permissão de acesso ao cadastro biblioteca
	ac_usuario	boolean	1	Sim		Permissão de acesso ao cadastro de usuários
	ac_catalogo	boolean	1	Sim		Permissão de acesso ao cadastro de bibliografias
	ac_acervo	boolean	1	Sim		Permissão de acesso ao

					cadastro de acervo
	ac_emprestimo	boolean	1	Sim	Permissão de acesso cadastro de emprestimo
	ac_usuario	boolean	1	Sim	Permissão de acesso cadastro de usuários

Tabela 33: Índices da tabela usuario

Nome	Campos	Classificação
idx_usuario_login	login	Asc
idx_usuario_nome	nome	Asc

Tabela 34: Regras da tabela usuario

Nome	Campos	Regra
reg_sexo	sexo	“F” ou “M”

- **Calendário** – Tabela de apoio, para impedir que a data de devolução do acervo aconteça em domingos e feriados, como mostra a Tabela 35.

Tabela 35: Tabela calendario

	campo	tipo	tam	req	único	descrição
Pk	id_data	Data	8	Sim	Sim	Data
	descrição	Varchar	50	Sim		Descrição
	feriado	Bool	1			Feriado
	observacao	Text				Dados complementares

- **Emprestimo** – Principal tabela deste módulo, responsável pelo empréstimo do acervo, como mostra a Tabela 36.

Tabela 36: Tabela emprestimo

	campo	tipo	tam	req	único	descrição
Pk	id_emprestimo	Número-int4	4	Sim	Sim	Código seqüencial
	is_usuario	Número-int4	4	Sim		Usuário

is_acervo	Número-int4	2	Sim	Acervo emprestado
dtemprestimo	Data hora	8	Sim	Data e hora que ocorreu o empréstimo
dtdevolucao	Data	4	Sim	Data programada para a entrega do acervo
dtentrega	Data hora	8		Data e hora que ocorreu a entrega do acervo
multa	Moeda	8		Multa aplicada pelo atraso na entrega
is_func_emp	Número-int4	4	Sim	FK(usuario) – Usuário que realizou o empréstimo
is_func_ent	Número-int4	4		FK(usuario) – Usuário que recebeu a devolução

Tabela 37: Índices da tabela emprestimo

Nome	Campos	Classificação
idx_acervo	is_acervo	Asc
idx_usuario	is_usuario	Asc

Tabela 38: Regras da tabela emprestimo

Nome	Campos	Regra
reg_multa	multa	≥ 0
reg_dtdevolucao	dtdevolucao	\geq dtemprestimo
reg_dtentrega	dtentrega	\geq dtemprestimo

4.3 - Acesso multiusuário

O acesso ao sistema será feito através do *browser* do usuário e não através de diretórios compartilhados, como no sistema CDS/ISIS. Através deste recurso impede-se que o usuário possa alterar ou mesmo apagar estruturas essenciais ao sistema. O controle de conexão ao sistema deverá ser feito à parte

disso através de conexão TCP/IP, controlado através de um *firewall*¹² e do SGBD.

O controle de segurança SGBD será realizado através da autenticação de um usuário com restrições de acesso. As restrições serão no sentido de impedir a remoção de estruturas importantes do sistema através de comandos SQL, exemplo *drop table* ou mesmo *drop database*. Para evitar essa circunstância, os programas criados pelo PHP farão a conexão através de um usuário com restrições de acesso.

4.4 - Cópia de segurança e recuperação

O processo de cópia poderá ser realizado mesmo estando o sistema sendo utilizado pelos usuários, *backup on-line*, através do utilitário *pg_dump*.

O *backup on-line* é um recurso interessante oferecido pelo PostgreSQL, pois não haverá a necessidade de ficar com o sistema desabilitado enquanto se realiza a cópia, como acontece no CDS/ISIS. Uma outra vantagem refere-se em manter a sincronia com a base de dados, ou seja, a cópia realmente representa a base de dados em um determinado momento.

4.5 – Políticas de acesso

A política de acesso tem como objetivo atribuir a cada usuário as suas permissões de trabalho. As regras de permissões são estabelecidas pela tabela

¹² *Firewall* – Software responsável por filtrar pacotes, evitando o acesso externo.

usuario, através da definição de alguns campos, Tabela 39.

Tabela 39: Permissão de acesso

<i>is_biblioteca</i>	O usuário fará a manutenção dos cadastros pertencentes a biblioteca que estiver lotado. Esse campo é de preenchimento obrigatório e define o comportamento das permissões.
<i>ac_biblioteca</i>	Permissão de acesso ao cadastro biblioteca.
<i>ac_usuario</i>	Permissão de acesso ao cadastro de usuários pertencentes à mesma biblioteca
<i>ac_catalogo</i>	Permissão de acesso ao módulo de catálogo bibliográfico
<i>ac_acervo</i>	Permissão de acesso ao módulo de acervos da biblioteca
<i>ac_emprestimo</i>	Permissão de acesso ao módulo de empréstimo e devolução
<i>ac_usuario</i>	Permissão de acesso ao cadastro de usuários
<i>ac_master</i>	Permissão de acesso pleno ao sistema

Na instalação do banco de dados, recomenda-se ao administrador do sistema a criação de uma biblioteca inicial e um usuário com acesso pleno. Para criar um usuário com acesso pleno preencha o campo *ac_master* com o valor *true*.

5 – CONCLUSÃO

O projeto de banco de dados proposto neste trabalho é uma alternativa de sistema para solucionar os problemas e limitações do aplicativo CDS/ISIS. Aproveitando a tecnologia do SGBD para reduzir o trabalho e a manutenibilidade originadas pelo desenvolvimento do aplicativo.

O *script* de criação de bancos de dados disponibilizado, anexo A, está em conformidade com o padrão SQL 92. Essa conformidade torna possível a adaptação para outros bancos de dados relacionais, comerciais ou não.

O trabalho atual restringe às definições das bases de dados relacionadas ao empréstimo do acervo. Um trabalho futuro refere-se à montagem da rotina de reservas de acervo para a solicitação de empréstimos. O módulo de reservas não foi proposto nesse trabalho.

Uma opção é utilizar a interface gráfica para que o sistema não dependa do servidor Web. Esta é uma implementação conveniente para que o aplicativo possa ser utilizado por bibliotecas particulares ou mesmo pequenas bibliotecas.

A implementação do aplicativo em linguagem PHP faz parte de outro trabalho realizado por outro aluno, o qual detalhará toda a funcionalidade do sistema através de *use cases* e implementação do aplicativo. A implementação está em um estado adiantado de desenvolvimento, disponibilizando menus, cadastros básicos e rotina de empréstimo. A tela principal do Projeto Biblios corresponde à Figura 9.

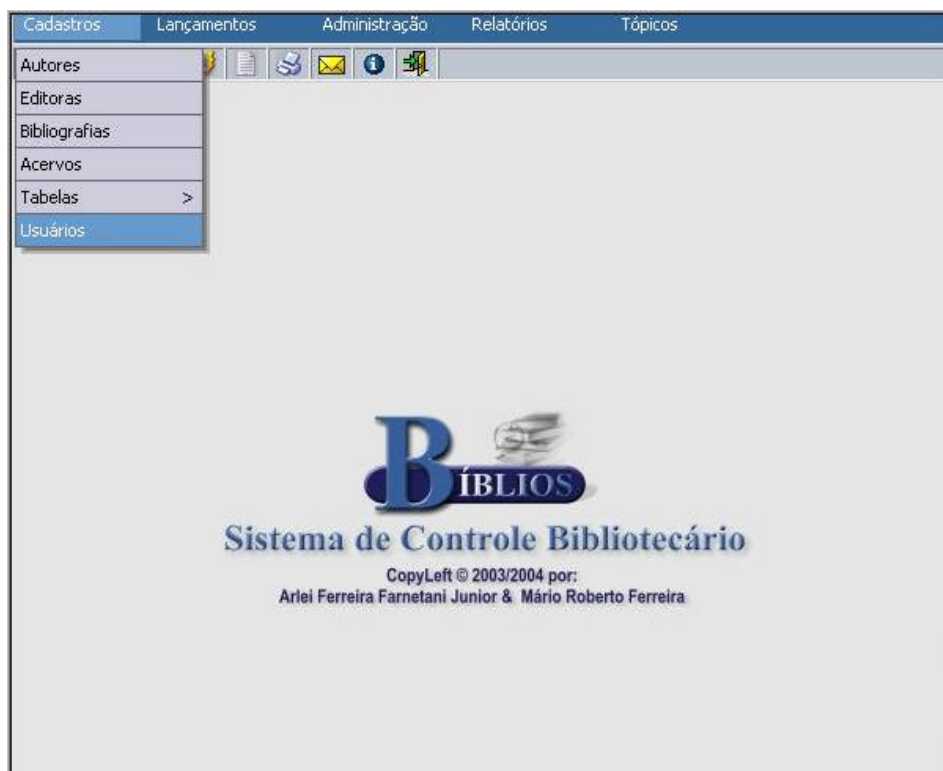


Figura 9: Tela principal do Projeto Biblios

6 - REFERÊNCIAS BIBLIOGRÁFICAS

[Apache (2003)] Servidor Web Apache. [on-line]. Disponível na Internet via
www url: <http://www.apache.org>. Arquivo capturado em 10 de
outubro de 2003.

[BIREME (2003)] Bireme. [on-line]. Disponível na Internet via www url:
<http://www.bireme.br>. Arquivo capturado em 15 de
outubro de 2003.

[BSD (2003)] *Berkeley Software Design*. [on-line]. Disponível na Internet via
www url: <http://www.bsd.org/>. Arquivo capturado em 20
de outubro de 2003.

[Date (2000)] DATE, C. J. *Introdução à Sistemas de Banco de Dados*. 7a. ed.
Rio de Janeiro: Campus, 2000.

[DIA (2003)] Lysoft. [on-line]. Disponível na Internet via www url:
<http://www.lysoft.org>. Arquivo capturado em 10 de
outubro de 2003.

[GNU (2003)] *Free Software Foundation. GNU's Not Unix!*. [on-line].
Disponível na Internet via www. url: <http://www.gnu.org/>.
Arquivo capturado em 10 de outubro de 2003.

[Gnuteca (2004)] Gnuteca. [on-line]. Disponível na Internet via www. url:
<http://gnuteca.codigolivre.org.br>. Arquivo capturado
em 05 de janeiro de 2004.

- [ISIS (2003)] UNESCO. *The UNESCO micro CDS/ISIS software*. [on-line].
Disponível na Internet via www. url: <http://www.unesco.org/webworld/isis/isis.htm>. Arquivo capturado em 02 de outubro de 2003.
- [ISO 9126 (1991)] *INTERNATIONAL Organization for Standardization - ISO; International Electrotechnical Commission - IEC. ISO/IEC 9126*.
Genebra: ISO/IEC. 1ª ed. 1991.
- [LIMA (1999)] LIMA, Gercina Ângela Borém. *Softwares para automação de bibliotecas e centros de documentação na literatura brasileira até 1998*. Ci. Inf., set./dez. 1999, vol.28, no.3, p.310-321. ISSN 0100-1965.
- [Manzanos (2003)] MANZANOS, Norberto. *História del MicroIgis*. [on-line].
Disponível na Internet via www. url: <http://www.cnea.gov.ar/cac/ci/isis/manual/>. Arquivo capturado em 08 de novembro de 2003.
- [Oráculo (2004)] Oráculo. [on-line]. Disponível na Internet via www. url:
<http://www.oraculo.org.br>. Arquivo capturado em 10 de janeiro de 2004.
- [PHP (2003)] PHP. [on-line]. Disponível na Internet via www. url:
<http://www.php.net>. Arquivo capturado em 15 de outubro de 2003.
- [PostgreSQL (2003)] PostgreSQL Inc. PostgreSQL. [on-line]. Disponível na Internet via www url: <http://www.postgresql.org>.
Arquivo capturado em 10 de outubro de 2003.
- [Ramalho (1999)] RAMALHO, J. *SQL: a linguagem dos bancos de dados*. São Paulo, Berkeley, 1999.

[Uchôa (2003)], Kátia Cilene Amaral. Introdução à Cibercultura / Kátia Cilene Amaral Uchôa - 3.ed. Lavras: UFLA/FAEPE, 2003. 114 p. : il. - Curso de Pós-Graduação “Lato Sensu” (Especialização) a Distância: Administração em Redes Linux, 2003.

[UFOP (2004)] UFOP - Universidade Federal de Ouro Preto. *SISBIN - Sistema de Bibliotecas e Informação*. [on-line]. Disponível na Internet via www. url: <http://www.ufop.br>. Arquivo capturado em 10 de janeiro de 2004.

[UFPE (2004)] UFPE - Universidade Federal de Pernambuco. *SIB - Sistemas de Bibliotecas da UFPE*. [on-line]. Disponível na Internet via www. url: <http://www.ufpe.br/>. Arquivo capturado em 11 de janeiro de 2004.

ANEXO A - Esquema banco de dados Biblios

```
--  
-- PostgreSQL database dump  
--  
  
SET SESSION AUTHORIZATION 'postgres';  
  
CREATE DATABASE biblios;  
  
REVOKE ALL ON SCHEMA public FROM PUBLIC;  
GRANT ALL ON SCHEMA public TO PUBLIC;  
  
SET search_path = public, pg_catalog;  
  
CREATE SEQUENCE seq_autor  
  START WITH 1  
  INCREMENT BY 1  
  MAXVALUE 2147483647  
  NO MINVALUE  
  CACHE 1;  
  
CREATE SEQUENCE seq_acervo  
  INCREMENT BY 1  
  MAXVALUE 2147483647  
  NO MINVALUE  
  CACHE 1;  
  
CREATE SEQUENCE seq_assunto  
  START WITH 1  
  INCREMENT BY 1  
  MAXVALUE 2147483647  
  NO MINVALUE  
  CACHE 1;  
  
CREATE SEQUENCE seq_biblioteca  
  START WITH 1  
  INCREMENT BY 1  
  MAXVALUE 2147483647  
  NO MINVALUE  
  CACHE 1;  
  
CREATE SEQUENCE seq_categoria  
  START WITH 1  
  INCREMENT BY 1  
  MAXVALUE 2147483647  
  NO MINVALUE  
  CACHE 1;
```

```
CREATE SEQUENCE seq_tipobiblio
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;
```

```
CREATE SEQUENCE seq_curso
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;
```

```
CREATE SEQUENCE seq_editora
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;
```

```
CREATE SEQUENCE seq_idioma
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;
```

```
CREATE SEQUENCE seq_listabiblio
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;
```

```
CREATE SEQUENCE seq_tipousuario
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;
```

```
CREATE SEQUENCE seq_usuario
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 2147483647
  NO MINVALUE
  CACHE 1;
```

```
CREATE SEQUENCE seq_emprestimo
  START WITH 1
```

```

INCREMENT BY 1
MAXVALUE 2147483647
NO MINVALUE
CACHE 1;

CREATE SEQUENCE seq_listabibliopalavra
INCREMENT BY 1
MAXVALUE 2147483647
NO MINVALUE
CACHE 1;

CREATE TABLE acervo (
  id_acervo integer DEFAULT nextval('seq_acervo'::text) NOT NULL,
  is_listabiblio integer NOT NULL,
  tomo character varying(15),
  dttomo date,
  tipoaquisicao smallint NOT NULL,
  dtbaixa date,
  tipobaixa smallint,
  status smallint NOT NULL,
  dados character varying(70),
  codigo character varying(20),
  is_biblioteca integer NOT NULL,
  CONSTRAINT reg_status CHECK (((status >= 1) AND (status <= 6))),
  CONSTRAINT reg_tipoaquisicao CHECK (((tipoaquisicao >= 1) AND (tipoaquisicao <= 3))),
  CONSTRAINT reg_tipobaixa CHECK (((tipobaixa >= 1) AND (tipobaixa <= 3)))
);

CREATE TABLE assunto (
  id_assunto integer DEFAULT nextval('seq_assunto'::text) NOT NULL,
  assunto character varying(100) NOT NULL
);

CREATE TABLE autor (
  id_autor integer DEFAULT nextval('seq_autor'::text) NOT NULL,
  autor character varying(100) NOT NULL,
  apelido character varying(50)
);

CREATE TABLE biblioteca (
  id_biblioteca integer DEFAULT nextval('seq_biblioteca'::text) NOT NULL,
  biblioteca character varying(100) NOT NULL,
  is_usuario integer,
  telefone1 character varying(15),
  telefone2 character varying(15)
);

CREATE TABLE calendario (
  id_data date NOT NULL,
  descricao character varying(50) NOT NULL,
  feriado boolean,

```



```

observacao text
);

CREATE TABLE categoria (
  id_categoria integer DEFAULT nextval('seq_categoria)::text) NOT NULL,
  categoria character varying(100) NOT NULL
);

CREATE TABLE tipobiblio (
  id_tipobiblio smallint DEFAULT nextval('seq_tipobiblio)::text) NOT NULL,
  tipobiblio character varying(100) NOT NULL
);

CREATE TABLE curso (
  id_curso integer DEFAULT nextval('seq_curso)::text) NOT NULL,
  curso character varying(50) NOT NULL
);

CREATE TABLE editora (
  id_editora integer DEFAULT nextval('seq_editora)::text) NOT NULL,
  nomecompleto character varying(100) NOT NULL,
  nomeabreviado character varying(100) NOT NULL,
  endereco character varying(50),
  complemento character varying(40),
  cep character varying(9),
  cidade character varying(40),
  uf character(2),
  pais character varying(40),
  telefone character varying(15),
  fax character varying(15),
  site character varying(100),
  email character varying(60),
  observacao text
);

CREATE TABLE emprestimo (
  id_emprestimo integer DEFAULT nextval('seq_emprestimo)::text) NOT NULL,
  is_usuario integer NOT NULL,
  is_acervo integer NOT NULL,
  dtemprestimo timestamp without time zone DEFAULT ('now)::text)::timestamp(6) with time
zone NOT NULL,
  dtdevolucao date NOT NULL,
  dtentrega timestamp with time zone,
  multa money,
  is_func_emp integer NOT NULL,
  is_func_dev integer,
  CONSTRAINT reg_dtdevolucao CHECK (((dtdevolucao)::timestamp without time zone >=
dtemprestimo)),
  CONSTRAINT reg_dtentrega CHECK ((dtentrega >= (dtemprestimo)::timestamp with time
zone)),
  CONSTRAINT reg_multa CHECK ((multa >= '$0.00)::money))

```

```

);

CREATE TABLE idioma (
  id_idioma integer DEFAULT nextval('seq_idioma)::text) NOT NULL,
  idioma character varying(100) NOT NULL
);

CREATE TABLE listabiblio (
  id_listabiblio integer DEFAULT nextval('seq_listabiblio)::text) NOT NULL,
  titulo character varying NOT NULL,
  subtitulo character varying,
  isbn_issn character varying(30),
  is_categoria integer NOT NULL,
  is_editora integer,
  cutter character varying(30),
  cdd character varying(30),
  cdu character varying(30),
  volume character varying(30),
  numero character varying(20),
  edicao character varying(20),
  capitulos smallint,
  qtdepag smallint,
  tipobiblio smallint NOT NULL,
  is_idioma integer NOT NULL,
  localpublicacao character varying(70),
  abstract text,
  anopublicacao smallint NOT NULL,
  conteudo text,
  tempo time without time zone,
);

CREATE TABLE listabiblioassunto (
  is_listabiblio integer NOT NULL,
  is_assunto integer NOT NULL
);

CREATE TABLE listabiblioautor (
  is_listabiblio integer NOT NULL,
  is_autor integer NOT NULL,
  tipoautor smallint NOT NULL,
  CONSTRAINT reg_tipoautor CHECK (((tipoautor >= 1) AND (tipoautor <= 3)))
);

CREATE TABLE listabibliopalavra (
  id_palavra integer DEFAULT nextval('seq_listabibliopalavra)::text) NOT NULL,
  is_listabiblio integer NOT NULL,
  palavra character varying(100) NOT NULL
);

CREATE TABLE tipousuario (
  id_tipousuario integer DEFAULT nextval('seq_tipousuario)::text) NOT NULL,

```

```

tipo character varying(50) NOT NULL,
qtdedias smallint NOT NULL,
qtdeacervo smallint NOT NULL,
CONSTRAINT reg_dias CHECK ((qtdedias > 0)),
CONSTRAINT reg_qtde CHECK ((qtdeacervo > 0))
);

CREATE TABLE usuario (
id_usuario integer DEFAULT nextval('seq_usuario)::text) NOT NULL,
is_tipousuario integer NOT NULL,
nome character varying(100) NOT NULL,
endereco character varying(100),
complemento character varying(40),
bairro character varying(50),
cidade character varying(50),
uf character(2),
cep character varying(8),
telefone character varying(15),
celular character varying(15),
dtnasc date,
sexo character(1),
email character varying(60),
is_curso integer,
matricula character varying(30),
cpf character varying(11),
rg character varying(20),
orgexped character varying(15),
bloqueado boolean DEFAULT false,
motivobloq text,
login character varying(15) NOT NULL,
senha character varying(60),
dtinclusao date DEFAULT ('now)::text)::date NOT NULL,
dtbloq date,
CONSTRAINT reg_sexo CHECK (((sexo = 'F'::bpchar) OR (sexo = 'M'::bpchar)))
);

CREATE UNIQUE INDEX idxu_autor ON autor USING btree (autor);

CREATE UNIQUE INDEX idxu_curso ON curso USING btree (curso);

CREATE UNIQUE INDEX idxu_tipobiblio ON tipobiblio USING btree (tipobiblio);

CREATE UNIQUE INDEX idxu_nomecompleto ON editora USING btree (nomecompleto);

CREATE UNIQUE INDEX idxu_nomeabreviado ON editora USING btree (nomeabreviado);

CREATE UNIQUE INDEX idxu_idioma ON idioma USING btree (idioma);

CREATE INDEX idx_tombo ON acervo USING btree (tombo);

CREATE UNIQUE INDEX idxu_assunto ON assunto USING btree (assunto);

```

```

CREATE UNIQUE INDEX idxu_biblioteca ON biblioteca USING btree (biblioteca);

CREATE UNIQUE INDEX idxu_categoria ON categoria USING btree (categoria);

CREATE UNIQUE INDEX idxu_listabiblio_palavra ON listabibliopalavra USING btree
(is_listabiblio, palavra);

CREATE UNIQUE INDEX idxu_tipousuario_tipo ON tipousuario USING btree (tipo);

CREATE INDEX idx_usuario_nome ON usuario USING btree (nome);

CREATE UNIQUE INDEX idxu_usuario_login ON usuario USING btree (login);

CREATE INDEX idx_apelido ON autor USING btree (apelido);

CREATE INDEX idx_cdd ON listabiblio USING btree (cdd);

CREATE INDEX idx_cdu ON listabiblio USING btree (cdu);

CREATE INDEX idx_isbn_issn ON listabiblio USING btree (isbn_issn);

CREATE INDEX idx_subtitulo ON listabiblio USING btree (subtitulo);

CREATE INDEX idx_cutter ON listabiblio USING btree (cutter);

CREATE INDEX idx_titulo ON listabiblio USING btree (titulo);

CREATE INDEX idx_usuario ON emprestimo USING btree (is_usuario);

CREATE INDEX idx_acervo ON emprestimo USING btree (is_acervo);

ALTER TABLE ONLY acervo
  ADD CONSTRAINT acervo_pkey PRIMARY KEY (id_acervo);

ALTER TABLE ONLY assunto
  ADD CONSTRAINT assunto_pkey PRIMARY KEY (id_assunto);

ALTER TABLE ONLY autor
  ADD CONSTRAINT autor_pkey PRIMARY KEY (id_autor);

ALTER TABLE ONLY biblioteca
  ADD CONSTRAINT biblioteca_pkey PRIMARY KEY (id_biblioteca);

ALTER TABLE ONLY calendario
  ADD CONSTRAINT calendario_pkey PRIMARY KEY (id_data);

ALTER TABLE ONLY categoria
  ADD CONSTRAINT categoria_pkey PRIMARY KEY (id_categoria);

ALTER TABLE ONLY curso

```

```

ADD CONSTRAINT curso_pkey PRIMARY KEY (id_curso);

ALTER TABLE ONLY editora
  ADD CONSTRAINT editora_pkey PRIMARY KEY (id_editora);

ALTER TABLE ONLY emprestimo
  ADD CONSTRAINT emprestimo_pkey PRIMARY KEY (id_emprestimo);

ALTER TABLE ONLY idioma
  ADD CONSTRAINT idioma_pkey PRIMARY KEY (id_idioma);

ALTER TABLE ONLY listabiblio
  ADD CONSTRAINT listabiblio_pkey PRIMARY KEY (id_listabiblio);

ALTER TABLE ONLY listabiblioassunto
  ADD CONSTRAINT listabiblioassunto_pkey PRIMARY KEY (is_listabiblio, is_assunto);

ALTER TABLE ONLY listabiblioautor
  ADD CONSTRAINT listabiblioautor_pkey PRIMARY KEY (is_listabiblio, is_autor);

ALTER TABLE ONLY listabibliopalavra
  ADD CONSTRAINT listabibliopalavra_pkey PRIMARY KEY (id_palavra);

ALTER TABLE ONLY tipousuario
  ADD CONSTRAINT tipousuario_pkey PRIMARY KEY (id_tipousuario);

ALTER TABLE ONLY usuario
  ADD CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario);

ALTER TABLE ONLY listabibliopalavra
  ADD CONSTRAINT fk_listabiblio1 FOREIGN KEY (is_listabiblio) REFERENCES
listabiblio(id_listabiblio) ON DELETE CASCADE;

ALTER TABLE ONLY listabiblio
  ADD CONSTRAINT fk_idioma FOREIGN KEY (is_idioma) REFERENCES
idioma(id_idioma);

ALTER TABLE ONLY listabiblio
  ADD CONSTRAINT fk_categoria FOREIGN KEY (is_categoria) REFERENCES
categoria(id_categoria);

ALTER TABLE ONLY listabiblio
  ADD CONSTRAINT fk_editora FOREIGN KEY (is_editora) REFERENCES
editora(id_editora);

ALTER TABLE ONLY listabiblioautor
  ADD CONSTRAINT fk_listabiblio2 FOREIGN KEY (is_listabiblio) REFERENCES
listabiblio(id_listabiblio) ON DELETE CASCADE;

ALTER TABLE ONLY listabiblioassunto

```

```

    ADD CONSTRAINT fk_listabiblio3 FOREIGN KEY (is_listabiblio) REFERENCES
listabiblio(id_listabiblio) ON DELETE CASCADE;

ALTER TABLE ONLY listabiblioassunto
    ADD CONSTRAINT fk_assunto FOREIGN KEY (is_assunto) REFERENCES
assunto(id_assunto);

ALTER TABLE ONLY listabiblioautor
    ADD CONSTRAINT fk_autor FOREIGN KEY (is_autor) REFERENCES autor(id_autor);

ALTER TABLE ONLY acervo
    ADD CONSTRAINT fk_listabiblio4 FOREIGN KEY (is_listabiblio) REFERENCES
listabiblio(id_listabiblio);

ALTER TABLE ONLY acervo
    ADD CONSTRAINT fk_biblioteca FOREIGN KEY (is_biblioteca) REFERENCES
biblioteca(id_biblioteca);

ALTER TABLE ONLY biblioteca
    ADD CONSTRAINT fg_usuario 1 FOREIGN KEY (is_usuario) REFERENCES
usuario(id_usuario);

ALTER TABLE ONLY emprestimo
    ADD CONSTRAINT fg_acervo FOREIGN KEY (is_acervo) REFERENCES
acervo(id_acervo);

ALTER TABLE ONLY emprestimo
    ADD CONSTRAINT fg_usuario FOREIGN KEY (is_usuario) REFERENCES
usuario(id_usuario);

ALTER TABLE ONLY usuario
    ADD CONSTRAINT fg_curso FOREIGN KEY (is_curso) REFERENCES curso(id_curso);

ALTER TABLE ONLY usuario
    ADD CONSTRAINT fg_tipousuario FOREIGN KEY (is_tipousuario) REFERENCES
tipousuario(id_tipousuario);

ALTER TABLE ONLY emprestimo
    ADD CONSTRAINT fg_func_emp FOREIGN KEY (is_func_emp) REFERENCES
usuario(id_usuario);

ALTER TABLE ONLY emprestimo
    ADD CONSTRAINT fg_func_dev FOREIGN KEY (is_func_dev) REFERENCES
usuario(id_usuario);

ALTER TABLE ONLY usuario
    ADD CONSTRAINT "fg_biblioteca" FOREIGN KEY (is_biblioteca) REFERENCES
biblioteca(id_biblioteca);

COMMENT ON SCHEMA public IS 'Standard public schema';

```

Anexo B - Lista de Abreviaturas

BSD - *Berkeley Software Design*

GB – Gibabyte (1024 mega bytes)

GPL – *General Public License* – Licença Pública Geral

MB – Megabyte (1024 bytes)

PHP – Hypertext Preprocessor

SGBD – *Software Gerenciador de Bancos de Dados*

SQL - *Structured Query Language*

TB – Terabyte (1024 giga bytes)

UML - *Unified Modeling Language*

UNESCO – Organização das Nações Unidas para a Educação, a Ciência e a Cultura