

MODELAGEM DE DADOS PARA UM SISTEMA DE GERENCIAMENTO DE ACERVOS PARA BIBLIOTECA ACADÊMICA

Autor(es): Charlie Bellow de Oliveira Pimentel Silva¹

Tutor externo: Caio Steglich Borges²

RESUMO

Este artigo objetiva-se na necessidade de implementação de um sistema de banco de dados de uma biblioteca acadêmica. Em razão do conhecimento incipiente do autor, esse estudo se faz necessário como suporte de estudo e aprendizado para aplicabilidade prática do autor posteriormente, bem como na importância de manter um sistema de gerenciamento de banco de dados eficaz para a Universidade Federal de Alagoas (UFAL). Foram utilizadas as tecnologias MySQL e Microsoft WorkBench para confecção do banco relacional, um dos modelos e linguagem mais utilizados no mercado e o brModelo para montar os modelos lógico e conceitual. O sistema visa cadastrar os usuários, os autores, o acervo disponível nas bibliotecas da universidade que permitirá criar eventos cadastrando datas de empréstimos e devoluções de acervo, cadastrando multas quando necessário, e contabilizando visitas a acervos online como periódicos.

- Palavras-chave: MySQL; banco de dados; biblioteca; acervo

INTRODUÇÃO

MOTIVO DA ESCOLHA DO OBJETO DE ESTUDO

A escolha deste objeto de estudo decorre de uma necessidade observada no contexto da Universidade Federal de Alagoas (UFAL) onde o autor realiza estágio na no setor de Desenvolvimento de Sistemas de Informação. Identificamos, através do relato do diretor do setor da biblioteca da universidade, que a guarda dos dados a respeito do acervo da biblioteca estava comprometida, pois alguns periódicos não estavam sendo encontrados no sistema atual. Portanto, foi identificada a necessidade de atualizar o banco de dados de forma a dirimir possíveis erros como estes.

¹ Acadêmico do Curso de Sistemas de Informação 3ª Semestre; E-mail: 6793114@aluno.uniassevi.com.br

² Tutor Externo do Curso de Sistemas de Informação – Polo Rio Grande do Sul; E-mail: nao-informado@tutor.uniassevi.com.br

Nesse sentido, observando a necessidade de implementação desse novo banco e o conhecimento incipiente do autor, esse artigo tem como objetivo servir de suporte e aprendizado sobre o tema para que o autor possa implementar a solução de forma viável e eficiente futuramente.

ESTRATÉGIAS DE ANÁLISE DO OBJETO

Para embasar o desenvolvimento do projeto foi adotada a abordagem de revisão bibliográfica em artigos e periódicos acadêmicos. Baseando-se em autores como Vergueiro e Carvalho (2000) que discutem o papel da biblioteca e sua coleta de dados, bem como os indicadores de qualidade das bibliotecas em universidades públicas; e Masiero (2001) que exploram as principais decisões e o processo organizacional utilizado no processo de criação de banco de dados para a biblioteca da Universidade Federal de São Paulo, visando facilitar a sua reprodução em outras universidades do país, acredita-se que seja possível respaldar o trabalho executado. Assim, esta pesquisa incluiu artigos científicos e relatórios técnicos focados no uso de tecnologia de criação de banco de dados em organizações públicas voltados ao contexto da biblioteca. Portanto, pode-se dizer que foi realizado um levantamento literário abrangente e pertinente com a temática em questão.

De acordo com o levantamento bibliográfico, partimos do pressuposto de que as bibliotecas precisam apresentar métodos gerenciais e práticas de trabalho que tenham uma resposta rápida e eficiente em relação às demandas da sociedade e às necessidades dos usuários (VERGUEIRO; CARVALHO, 2000 apud FILL et. al., 2006, p. 234). A coleta de dados se faz imprescindível no processo pois “atende a uma demanda institucional e serve de base para a distribuição de recursos orçamentários às bibliotecas e a outras finalidades internas, além de fornecer dados às publicações oficiais da Universidade” (FILL et. al., 2006, p. 234). Dessa forma, após a coleta, observa-se a necessidade de padronização dos dados coletados e, principalmente, da forma de coleta, buscando evitar ou minimizar erros de interpretação e dados que não correspondam à realidade.

Nesse sentido podemos dizer que quando implementado, o banco de dados da biblioteca torna-se “um instrumento dinâmico que permite às bibliotecas efetuarem, a qualquer momento, controles estatísticos individuais e acompanharem as informações cadastradas” (FILL et. al., 2006, p. 234) permitindo a realização de previsões e tomada de decisão que é, conforme Morettin e Bussab (2003), um aspecto importante da modelagem de dados.

Assim, considerando a quantidade extensa de dados que precisam ser gerenciados em uma biblioteca, é possível concordar com a afirmação de Fox e Marchionini (2001) de que as bibliotecas digitais envolvem uma integração de sistemas complexos, que inclui uma coleção de documentos com várias estruturas de mídias e conteúdos que misturam componentes de hardware e software interligados, evidenciando a complexidade desse estudo de caso. Entretanto, o que está sendo proposto aqui é uma solução bem mais simples para fins didáticos.

Diante do exposto, justifica-se a importância e a necessidade de um padrão de coleta de dados que reflita os interesses da instituição para que possa servir, efetivamente, como um sistema eficaz no gerenciamento do acervo universitário. Tal coleta foi realizada com base no sistema atual e na observação do próprio autor sobre a dinâmica da biblioteca. Como se trata de uma propriedade pública e sigilosa não pode ser exposta neste trabalho. Porém, todos os dados inseridos no modelo do banco proposto neste trabalho refletem as necessidades da biblioteca com algumas adaptações para que o projeto não se torne extenso e fuja do escopo proposto.

Dessa forma a modelagem de dados relacional é a abordagem escolhida devido a sua robustez e eficiência em fornecer relações complexas de forma estruturada e intuitiva objetivando facilitar a manipulação e recuperação de dados, ou seja, a modelagem relacional permite a criação de um esquema de banco de dados organizando as informações em tabelas com linhas e colunas claramente definidas (Codd, 1970). Assim, o banco de dados relacional pode garantir a integridade e consistência dos dados o que é imprescindível para sistemas que lidam com dados variados e de interesse público como é o caso das universidades, além de facilitar a implementação de consultas SQL proporcionando a flexibilidade e interação com o banco de dados.

CONSIDERAÇÕES CRÍTICAS E CRIATIVAS

Inicialmente faz-se necessário explicar o processo de conceituação do banco de dados, listando seus requisitos e o escopo do projeto conforme a observação realizada. Nesse sentido, de acordo com a revisão bibliográfica, conforme (Fill et. al., 2006, p. 238-239), e visando as necessidades da biblioteca da UFAL, verificou-se que os dados coletados pelas bibliotecas públicas, de modo geral, são:

- Identificação da biblioteca – nome do responsável pela biblioteca, nome da Biblioteca, registro no CRB/8, endereço, telefone e e-mail.

- Horário de funcionamento – indicação do horário em que a biblioteca permanece aberta ao público no período letivo e nas férias escolares.
- Acervo – indicação de itens incorporados ao acervo de livros, teses, títulos e fascículos de periódicos, multimeios e outros tipos de materiais com especificação do número de materiais processados no ano, baixas efetuadas.
- Aquisição – indicação de itens do acervo adquiridos por compra, permuta e doação de livros, teses, títulos de periódicos correntes, multimeios e outros tipos de materiais, subdivididos por tipo de procedência (nacional ou internacional).
- Usuários – número de usuários inscritos e potenciais da Universidade (docentes/pesquisadores, discentes de graduação, discentes de pós-graduação e outros usuários) bem como suas informações pessoais como nome, telefone, data de nascimento, email, curso, gênero, entre outros.
- Frequência de usuários – número de usuários que frequentaram a Biblioteca e número de consultas de livros e periódicos.
- Circulação – número de empréstimos, consultas, empréstimos entre bibliotecas (como biblioteca fornecedora e biblioteca solicitante) e devoluções.

Esses dados foram adaptados para compor um sistema mais simples, e a solução proposta para esse estudo visa atender requisitos funcionais menos complexos. Sommerville (2007) afirma que os requisitos funcionais objetivam especificar as ações que o sistema deve executar. Eles dependem da finalidade do software, do perfil dos usuários finais e da metodologia adotada pela organização para formular esses requisitos. Os requisitos funcionais pensados para este trabalho serão descritos abaixo de forma subjetiva, permitindo a compreensão geral do escopo do sistema proposto:

- RF01: O sistema deve permitir o cadastro de usuário utilizando informações básicas (nome, telefone, e-mail, curso, gênero, telefone, data de nascimento, entre outros);
- RF02: O sistema deve permitir que o bibliotecário faça o cadastro de um evento de empréstimo de um acervo identificando quem é o requerente de tal evento (uma outra biblioteca ou o próprio usuário);
- RF03: O sistema deve permitir a adição de novos usuários, novos acervos, novos autores, novos polos, novas bibliotecas, novos bibliotecários, novos cursos e novas multas;
- RF04: O bibliotecário pode publicar acervos online e estes acervos podem ser consultados publicamente de forma online;

- RF05: Uma multa deverá ser aplicada caso o usuário não devolva o acervo na data estipulada no momento da criação do evento;
- RF06: Uma biblioteca pode emprestar acervos para um usuário ou para uma biblioteca de outro polo;
- RF07: O sistema mostrará ao bibliotecário quando o usuário tiver uma multa pendente impedindo o empréstimo de acervos;
- RF08: Os usuários terão acesso aos acervos de todas as bibliotecas na página principal de consulta do sistema;

Com base nesses requisitos e para conceituar e explanar o modelo do banco de dados proposto para este estudo, foram criadas as seguintes modelagens: modelo Conceitual, modelo Lógico e modelo Físico. Cada uma delas será detalhada a seguir.

Modelo Conceitual

A modelagem conceitual de banco de dados auxilia no processo de organização e entendimento sobre os dados, demonstrando significativamente a aplicabilidade dos dados no contexto requerido pelos usuários. De acordo com Müller (2002) a modelagem de banco de dados também pode contribuir para dirimir a complexidade do projeto, fornecendo uma melhor compreensão e manipulação dos dados. A Figura 1 apresenta a modelagem do modelo conceitual de banco de dados conforme o requerido no escopo do trabalho.

Figura 1 - Modelo Conceitual

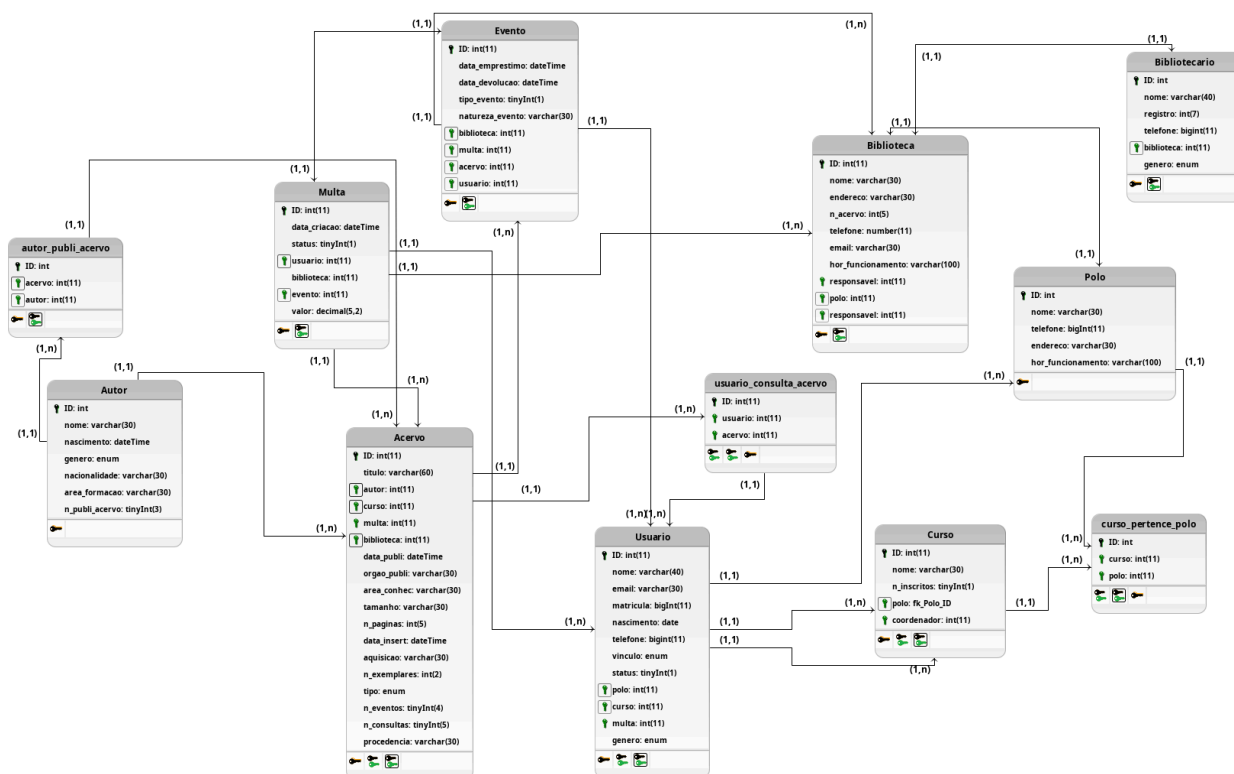
preciso entender que para cada item adicionado ao banco de dados é chamado de registro ou ocorrência e, segundo Heuser (2011, p. 37) a ocorrência (ou registro) é “o item específico da entidade ou do relacionamento (linha da tabela)”. Então, conforme o autor (idem), em um modelo relacional é fundamental que saibamos quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência através do relacionamento e a isso damos o nome de cardinalidade.

Conforme nossos estudos em sala de aula, pudemos observar que existem três tipos principais de cardinalidade, a saber: a) Um-para-um (1:1): Cada instância de uma entidade se relaciona com no máximo uma instância da outra entidade. Exemplificando seria: Um indivíduo possui apenas um CPF. b) Um-para-muitos (1:N): Uma instância de uma entidade pode se relacionar com várias instâncias da outra entidade, mas cada instância da outra entidade se relaciona com apenas uma da primeira. Como por exemplo: Um cliente pode fazer vários pedidos. c) Muitos-para-muitos (N:N): Cada instância de uma entidade pode se relacionar com várias instâncias da outra entidade, e vice-versa. Um exemplo seria: Um aluno pode se matricular em várias disciplinas, e uma disciplina pode ter vários alunos. As cardinalidades deste modelo proposto serão explanadas no detalhamento das tabelas do Modelo Lógico a seguir.

Modelo Lógico

O modelo apresentado na Figura 2 logo abaixo é um diagrama de Entidade-Relacionamento (ER) em sua representação lógica para um sistema de banco de dados projetado para gerenciar informações de um sistema de bibliotecas de universidades federais. Esse modelo traz informações mais detalhadas em relação ao modelo conceitual, porém mantendo a identificação da cardinalidade. As entidades, e alguns relacionamentos (quando necessário), do modelo conceitual é representado nesse modelo por tabelas. Os atributos são mais detalhados mostrando os tipos de seus respectivos valores. O código identificador é representado pela chave primária (primary key) na cor preta e a referência ao código identificador de outra entidade é representado pela chave estrangeira (foreign key) em verde. A seguir, pode-se observar a representação do modelo lógico:

Figura 2 - Modelo Lógico



De modo geral, pode-se observar que o modelo também foi realizado no BrModelo [2025?], as tabelas incluem: usuários, bibliotecas, bibliotecários, acervos, empréstimos, multas, entre outras. Cada tabela possui seus respectivos atributos, como id, nome, email para usuário e bibliotecários; e título, autor, entre outros, para acervos acompanhados de seus valores aceitáveis, como, varchar, int, dateTime, entre outros, além das chaves estrangeiras que materializam a relação entre as tabelas. Para facilitar o entendimento da ideia por trás do modelo lógico bem como as relações entre tabelas foi explicitado abaixo cada entidade onde cada palavra em **negrito** representa uma entidade que foi transformada em tabela:

- A universidade dispõe de vários cursos de graduação e pós-graduação (mestrado e doutorado), e de vários pólos espalhados pelo estado, onde cada polo possui uma **Biblioteca** própria e o **Evento** de empréstimo de acervos pode ser feito entre bibliotecas ou entre biblioteca e usuário. A **Biblioteca** pode ter várias unidades de um mesmo acervo de mesmo nome, autor, ano, entre outros, mas cada unidade do acervo deve ter um número de identificação único. A Biblioteca pode emprestar e solicitar empréstimo de acervos por meio de um **Evento**, bem como gerar **Multa**. Na **Biblioteca** deve ter o nome da

biblioteca, nome do responsável pela biblioteca (**Bibliotecário**), **Polo** ao qual pertence, quantidade de acervos, endereço, email para contato, telefone, horário de funcionamento, entre outros. A Cardinalidade seria: **Biblioteca-Evento** (1,n); **Biblioteca-Multa** (1,n); **Biblioteca-Polo** (1,1).

- O **Bibliotecário** deve possuir os dados: nome, registro do CRB, endereço, e-mail e a qual biblioteca é responsável. A Cardinalidade seria: **Bibliotecário-Biblioteca** (1,1).
- As informações sobre **Curso**, Alunos e Professores (**Usuários**), **Bibliotecas**, seus respectivos **Acervos** bem como a movimentação por meio de **Eventos** de empréstimos e aplicação de **Multa** são essenciais para a administração da Biblioteca. Cada **Curso** deve ter nome, os pólos no qual é ministrado, os professores responsáveis (coordenadores) e seus respectivos polos, a quantidade de inscritos em cada curso. E a quantidade de alunos inscritos por curso pode variar. A Cardinalidade é: **Curso-Polo** (n,n); **Curso-Usuário** (1,n).
- Para cada Aluno e/ou Professor (**Usuário**) existe uma ficha cadastral na biblioteca contendo nome, email, telefone, curso, tipo de vínculo (discente ou docente), **Polo** e status (ativo ou inativo). Cada **Usuário** pode consultar os **Acervos** disponíveis online como TCC, Teses e Dissertações; pode solicitar empréstimos de acervos solicitando um **Evento**, fazer devoluções de acervos, e pode pagar as **Multas**. A Cardinalidade fica sendo: **Usuário-Polo** (1,n); **Usuário-Multa** (1,n); **Usuário-Evento** (1,n); **Usuário-Curso** (1,n).
- No **Acervo**, cada acervo só pode ser emprestado a um aluno ou a uma biblioteca interna de outro pólo por vez. A **Biblioteca** pode cadastrar o empréstimo ou solicitar o empréstimo por meio de **Eventos**, ambas as ações podem ocorrer mais de uma vez e várias vezes, mas nunca em duplicidade ao mesmo tempo. No acervo deve conter o título, código identificador, autor, data de publicação, órgão publicador (revista, periódico, etc), número de **Multas** associadas a ele, número de **Eventos**, área de conhecimento, número de páginas, tamanho (em minutos ou centímetros ou megabytes (MB)), tipo (periódico ou Físico) curso ao qual está vinculado, **Biblioteca** a qual pertence, e número de exemplares (quantas unidades constam desse mesmo item), data de inserção no acervo, tipo de aquisição (compra, permuta e doação de livros) e procedência (internacional, nacional, outra Biblioteca, entre outros). Em caso de periódicos disponíveis online deve conter a quantidade de consultas que foram feitas. Assim, a Cardinalidade é: **Acervo-Biblioteca** (1,n); **Acervo-Multa** (1,n); **Acervo-Evento** (1,n).

- O **Evento** do mesmo acervo (com o mesmo número de identificação) só pode ocorrer depois do prazo final pré-determinado (de 7 dias) acabar e a devolução ser efetivada, caso a devolução não seja feita, o **Usuário** pagará uma **Multa** (3 reais + 0,50 a cada dia). No **Evento** deve conter o **Usuário** responsável, **Acervo**, **Biblioteca** responsável pelo empréstimo (onde o acervo está localizado) data de empréstimo, data de devolução prevista (ou data de devolução efetivada) e a **multa** se houver, tipo de empréstimo (entre bibliotecas ou para usuários). Logo, a Cardinalidade fica assim: **Evento-Biblioteca** (1,n); **Evento-Multa** (1,1); **Evento-Usuário** (1,n).
- A **Multa** só pode ser atribuída a uma pessoa e deve identificar quem é o usuário ou biblioteca responsável por ela e referenciar o **Evento** que a originou contendo data de criação para que seja contabilizado valor devido e dias de atraso, deve conter seu status (paga ou pendente). A **Multa** pode ser cadastrada por uma **Biblioteca** responsável. Já a Cardinalidade é: **Multa-Evento** (1,n); **Multa-Biblioteca** (1,n); **Multa-Usuário** (1,n).

Como alguns relacionamentos têm uma cardinalidade de muitos para muitos (n, n), então faz-se necessária a criação de tabelas separadas para facilitar a relação entre os dados. É o caso das tabelas:

- **Curso_pertence_polo**: um curso pode pertencer a vários pólos e um polo pode ter vários cursos. Nesse caso, a cardinalidade entre essa tabela e as tabelas **Curso** e **Polo** é de (1, n).
- **Usuario_consulta_acervo**: Um usuário pode consultar vários acervos e os acervos podem ser consultados por vários usuários. Nesse caso, a cardinalidade entre essa tabela e as tabelas **Usuário** e **Acervo** é de (1, n).
- **Autor_publi_acervo**: Um autor pode ter vários acervos publicados, e um acervo pode ter vários autores. Nesse caso, a cardinalidade entre essa tabela e as tabelas **Autor** e **Acervo** é de (1, n).

Seguem listados abaixo alguns relacionamentos entre as entidades que não foram contemplados na descrição acima:

- Nas universidades federais o aluno só pode está matriculado em um curso por vez, então, um usuário só pode estar matriculado em um único curso, e um curso pode ter vários alunos. Logo, a cardinalidade é de (1,1).

- Um usuário pode solicitar vários eventos, mas um evento só pode ter um usuário responsável por vez. Logo, a cardinalidade é de (1,n).
- Um curso pode pertencer a vários pólos e um polo pode ter vários cursos. Logo, a cardinalidade é de (1,n).
- Um autor pode ter vários acervos publicados, e um acervo pode ter vários autores. Logo, a cardinalidade é de (n,n).
- Uma multa pode ter apenas um usuário como responsável, e um usuário pode ser responsável por várias multas. Logo, a cardinalidade é de (1,n).
- Uma biblioteca só pode ter um bibliotecário responsável e um bibliotecário só pode ser responsável por uma biblioteca. Logo, a cardinalidade é de (1,1).

A integração dessas tabelas ajudam a dar vida e sentido ao sistema e permite uma melhor compreensão para a aplicabilidade da implementação de um sistema, conforme os requisitos levantados.

Modelo Físico

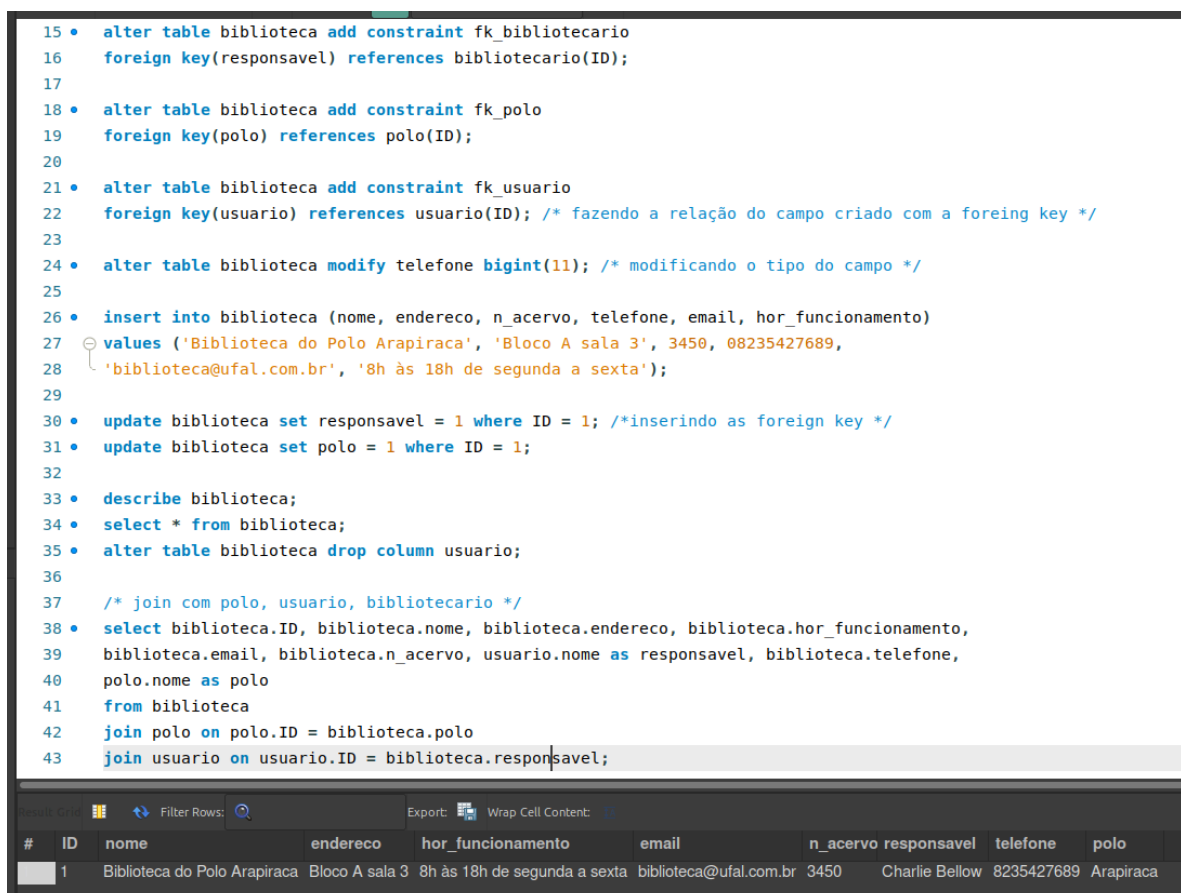
O modelo físico foi desenvolvido com um servidor local rodando no xampp na versão v.8.2.4-0 para linux, o código foi desenvolvido na linguagem SQL (Structured Query Language) e foi escrito no Microsoft Workbench [2025?] versão para linux. Portanto, foi desenvolvido um banco de dados MySQL que segundo Costa (2021, p. 15) é “um Banco de Dados relacional (RDBMS – Relational Database Management Systems) com um modelo de cliente-servidor” que possui código aberto que proporciona a criação e gerenciamento de bancos de dados baseados no modelo relacional. A escolha do MySQL (2007) se evidencia no fato de que o MySQL é extremamente rápido, armazena os dados em tabelas e é confiável e por isso, é indicado para sistemas de várias naturezas e tamanhos.

O modelo físico consiste na implementação de um sistema de banco de dados, que conforme Costa (2021, p. 21) “é um banco de dados por si só, se refere a um sistema computadorizado que pode possuir uma forma de acesso gráfico ou não, em linguagem natural, menus de acessos em SQL” que efetuam o armazenamento e a manutenção dos registros de dados. O autor compara os espaços disponíveis no banco de dados para armazenamento de registros com um “armário eletrônico de arquivamentos, ou seja, um repositório ou recipiente para uma coleção de arquivos de dados computadorizados, onde os usuários ou os programadores irão fazer as requisições de dados” (COSTA, 2021, p.14).

Nesse sentido, e atendendo à necessidade da biblioteca da UFAL, o banco de dados foi desenvolvido em plataforma de hardware com processador NVidia, sistema operacional Linux, distribuição Mint, Apache com SSL como servidor e MySQL para armazenamento de dados. A programação desta base permite ao bibliotecário efetuar a entrada de dados e, com isso, permitindo o acompanhamento das atividades executadas pelo departamento técnico ou pelas bibliotecas.

Abaixo segue alguns exemplos de tabelas implementadas e seus respectivos dados populados de forma a compor um seed (dados fictícios que servem de exemplo) para testes e demonstrações conforme figuras 3 a 9.

Figura 3 - Tabela Biblioteca - (Modelo Físico)



```

15 • alter table biblioteca add constraint fk_bibliotecario
16 foreign key(responsavel) references bibliotecario(ID);
17
18 • alter table biblioteca add constraint fk_polo
19 foreign key(polo) references polo(ID);
20
21 • alter table biblioteca add constraint fk_usuario
22 foreign key(usuario) references usuario(ID); /* fazendo a relação do campo criado com a foreign key */
23
24 • alter table biblioteca modify telefone bigint(11); /* modificando o tipo do campo */
25
26 • insert into biblioteca (nome, endereco, n_acervo, telefone, email, hor_funcionamento)
27 values ('Biblioteca do Polo Arapiraca', 'Bloco A sala 3', 3450, 08235427689,
28 'biblioteca@ufal.com.br', '8h às 18h de segunda a sexta');
29
30 • update biblioteca set responsavel = 1 where ID = 1; /*inserindo as foreign key */
31 • update biblioteca set polo = 1 where ID = 1;
32
33 • describe biblioteca;
34 • select * from biblioteca;
35 • alter table biblioteca drop column usuario;
36
37 /* join com polo, usuario, bibliotecario */
38 • select biblioteca.ID, biblioteca.nome, biblioteca.endereco, biblioteca.hor_funcionamento,
39 biblioteca.email, biblioteca.n_acervo, usuario.nome as responsavel, biblioteca.telefone,
40 polo.nome as polo
41 from biblioteca
42 join polo on polo.ID = biblioteca.polo
43 join usuario on usuario.ID = biblioteca.responsavel;

```

#	ID	nome	endereco	hor_funcionamento	email	n_acervo	responsavel	telefone	polo
1	1	Biblioteca do Polo Arapiraca	Bloco A sala 3	8h às 18h de segunda a sexta	biblioteca@ufal.com.br	3450	Charlie Bellow	8235427689	Arapiraca

Fonte: Elaborado pelo autor

Figura 4 - Tabela Acervo - (Modelo Físico)

```

25  /* adicionar a foreing key de autor, curso, multa, biblioteca */
26
27  • alter table acervo add constraint fk_acervo_autor
28    foreign key(autor) references autor(ID);
29
30  • alter table acervo add constraint fk_acervo_multa
31    foreign key (multa) references multa(ID);
32
33  • alter table acervo add constraint fk_acervo_biblioteca
34    foreign key (biblioteca) references biblioteca(ID);
35
36  • alter table acervo add constraint fk_acervo_curso
37    foreign key (curso) references curso(ID);
38
39  • insert into acervo (titulo, autor, curso, multa, biblioteca, data_publi, orgao_publi, area_conhec,
40    tamanho, n_paginas, data_insert, aquisicao, n_exemplares, tipo, procedencia)
41    values ('Anti-Édipo', 3, 1, 1, '2010-06-15 09:00:00', 'Editora 34', 'psicologia',
42    '20.45 X 18.6', 340, '2025-06-13', 'Compra', 1, 'Físico', 'Biblioteca de Maceió');
43
44  /* join com autor, curso, multa e biblioteca */
45
46  • select acervo.ID, acervo.titulo, autor.nome as autor, acervo.data_publi,
47    acervo.n_paginas, biblioteca.nome as biblioteca, acervo.n_consultas,
48    acervo.n_exemplares, acervo.area_conhec, acervo.orgao_publi,
49    curso.nome as curso, acervo.data_insert, acervo.procedencia, multa.ID as multa,
50    acervo.n_eventos, acervo.tamanho, acervo.tipo from acervo
51    join autor on autor.ID = acervo.autor
52    join curso on curso.ID = acervo.curso
53    join multa on multa.ID = acervo.multa

```

#	ID	titulo	autor	data_publi	n_paginas	biblioteca	n_consulta	n_exemplares	area_conhec	orgao_publ	curso	data_insert	procedencia	multa	n_eventos	tamanho	tipo
1		Anti-Édipo	Félix Guattari	2010-06-15 09:00:00	340	Biblioteca do Polo Ar...	0	1	psicologia	Editora 34	Psicologia	2025-06-13 00:00:00	Biblioteca de Maceió	1	0	20.45 X 18.6	Físico

Fonte: Elaborado pelo autor

Figura 5 - Tabela Usuário - (Modelo Físico)

```

30  • alter table usuario
31    modify column telefone bigint(11);
32
33  • insert into usuario (nome, email, matricula, telefone, vinculo, status, polo )
34    values ('Charlie Bellow', 'charlie@email.com', 15861245639, 829925467856, 'Discente', 1, 1);
35
36  • insert into usuario (nome, email, matricula, telefone, vinculo, status, polo )
37    values ('David Bernardo', 'david@email.com', 20861245435, 829945667885, 'Docente', 1, 1);
38
39  • alter table usuario add column nascimento date after matricula;
40
41  • update usuario set genero = 'HT' where ID = 1;
42  • update usuario set genero = 'M' where ID = 2;
43
44  • update usuario set nascimento = '1994-01-16' where ID = 1;
45  • update usuario set nascimento = '1985-05-05' where ID = 2;
46
47  • update usuario set curso = 1 where ID = 2;
48  • update usuario set multa = 1 where ID = 2;
49
50  /* join com polo, curso e multa */
51
52  • select usuario.ID, usuario.nome, usuario.genero,
53    usuario.email, usuario.matricula, usuario.nascimento, usuario.telefone,
54    usuario.vinculo, case when usuario.status = 1 then 'Ativo' when usuario.status = 0 then 'inativo' end as status,
55    polo.nome as polo, curso.nome as curso, multa.ID as ID_multa from usuario
56    join polo on polo.ID = usuario.polo
57    join curso on curso.ID = usuario.curso
58    join multa on multa.ID = usuario.multa order by nome;

```

#	ID	nome	genero	email	matricula	nascimento	telefone	vinculo	status	polo	curso	ID_multa
1		Charlie Bellow	HT	charlie@email...	15861245639	1994-01-16	829925467856	Discente	Ativo	Arapiraca	Psicologia	1
2		David Berna...	M	david@email.com	20861245435	1985-05-05	829945667885	Docente	Ativo	Arapiraca	Psicologia	1

Fonte: Elaborado pelo autor

Figura 6 - Tabela Evento - (Modelo Físico)

```

12
13 • alter table evento add column (biblioteca int, multa int, acervo int, usuario int);
14
15
16 • ALTER TABLE evento
17   ADD CONSTRAINT fk_evento_biblioteca
18   FOREIGN KEY (biblioteca) REFERENCES biblioteca(ID);
19
20 • alter table evento add constraint fk_evento_multa
21   foreign key (multa) references multa(ID);
22
23 • alter table evento add constraint fk_evento_usuario
24   foreign key (usuario) references usuario(ID);
25
26 • alter table evento add constraint fk_evento_acervo
27   foreign key (acervo) references acervo(ID);
28
29 • insert into evento (data_emprestimo, data_devolucao, tipo_evento, natureza_evento, biblioteca, multa, usuario)
30   values ('2025-06-12 09:00:00', 2025-06-21, 0, 'Emprestimo de livros', 1, 1, 1);
31
32 /* join com usuario, acervo, multa e biblioteca */
33
34 • Select evento.acervo, biblioteca.nome, evento.data_devolucao, evento.data_emprestimo, multa.ID as Id_multa, evento.natureza_evento,
35   case when evento.tipo_evento = 0 then 'Usuário' when evento.tipo_evento = 1 then 'Biblioteca' else '' end as tipo_evento, usuario.nome from evento
36   join biblioteca on evento.biblioteca = biblioteca.ID
37   join multa on multa.ID = evento.multa
38   join usuario on usuario.ID = evento.usuario;

```

#	acervo	nome	data_devolucao	data_emprestimo	Id_multa	natureza_evento	tipo_evento	nome
1		Biblioteca do Polo Arapiraca	2025-06-21 0...	2025-06-12 09:00:00	1	Emprestimo de livros	Usuário	Charlie Bellow

Fonte: Elaborado pelo autor

As tabelas que precisaram ser criadas para resolver a questão da cardinalidade, foram criadas relacionando as suas chaves estrangeiras, mas para que fizesse sentido para interpretação humana foi necessário fazer a junção entre tabelas utilizando o comando JOIN do SQL. Segue representação gráfica do código dessas junções:

Figura 7 - Tabela Usuario_consulta_acervo - (Modelo Físico)

```
1 • create table usuario_consulta_acervo (
2   ID int not null auto_increment,
3   usuario int(11),
4   acervo int(11),
5   primary key (ID),
6   foreign key (usuario) references usuario(ID),
7   foreign key (acervo) references acervo(ID)
8 ) default charset = utf8;
9
10 • describe usuario_consulta_acervo;
11 • select * from usuario_consulta_acervo;
12
13 • insert into usuario_consulta_acervo values (default, 1, 1);
14 • insert into usuario_consulta_acervo values (default, 2, 1);
15
16 • select usuario.ID, usuario.nome as usuario, acervo.titulo
17   from usuario join usuario_consulta_acervo
18   on usuario.ID = usuario_consulta_acervo.ID
19   join acervo on usuario_consulta_acervo.ID = acervo.ID;
```

Result Grid

#	ID	usuario	titulo
1		Charlie Bellow	Anti-Edipo

Fonte: Elaborado pelo autor

Figura 8 - Tabela Curso_pertence_polo - (Modelo Físico)

```

4  polo int(11),
5  primary key (ID),
6  foreign key (curso) references curso(ID),
7  foreign key (polo) references polo(ID)
8  ) default charset = utf8;
9
10 • describe curso_pertence_polo;
11 • select * from curso_pertence_polo;
12
13 • insert into curso_pertence_polo values (default, 1, 1);
14 • insert into curso_pertence_polo values (default, 1, 2);
15
16 • select curso.ID, curso.nome as curso, polo.nome as polo
17   from curso join curso_pertence_polo
18   on curso.ID = curso_pertence_polo.ID
19   join polo on curso_pertence_polo.ID = polo.ID;
20

```

#	ID	curso	polo
1		Psicologia	Arapiraca

Fonte: Elaborado pelo autor

Figura 9 - Tabela Autor_publico_Acervo - (Modelo Físico)

```

2  ID int not null auto_increment,
3  autor int(11),
4  acervo int(11),
5  primary key (ID),
6  foreign key (autor) references autor(ID),
7  foreign key (acervo) references acervo(ID)
8  ) default charset = utf8;
9
10 • describe autor_publico_acervo;
11 • select * from autor_publico_acervo;
12
13 • insert into autor_publico_acervo values (default, 1, 1);
14 • insert into autor_publico_acervo values (default, 2, 1);
15 • desc acervo;
16
17 • select autor.ID, autor.nome, autor.n_publico_acervo,
18   acervo.titulo, acervo.data_publico from autor
19   join autor_publico_acervo on autor.ID = autor_publico_acervo.ID
20   join acervo on acervo.ID = autor_publico_acervo.ID;

```

#	ID	nome	n_publico_acervo	titulo	data_publico
1		Gilles Deleuze	2	Anti-Édipo	2010-06-15 09:00:00

Fonte: Elaborado pelo autor

Por fim, os códigos e arquivos exportados do projeto podem ser encontrados no Github (site que usa o git para versionamento de código e salva-os na nuvem), no seguinte link: <https://github.com/CharlieBellow/biblioteca-uniassevi> para fins de consulta e avaliação.

REFLEXÕES FINAIS

O trabalho reforça a importância de adotar uma abordagem sistemática e tecnológica na resolução de problemas em contextos reais, em especial nas instituições públicas. Portanto, observou-se como a formação do acadêmico, autor deste texto foi impactada significativamente. Os desafios encontrados na execução deste trabalho foi a complexidade de integrar diversas fontes de dados e relacioná-las entre si de forma que faça sentido e abarque a real necessidade do contexto em questão. Porém, tais desafios foram superados com muito estudo aprofundado e foco na questão a ser resolvida. Apesar das dificuldades, este sistema pode ser desenvolvido.

Contudo, por se tratar de instrumento dinâmico de coleta de dados, ele pode estar sujeito a alterações futuras que reflitam as novas necessidades do Sistema considerando também os avanços na área de informática. Por outro lado, o desenvolvimento de uma base de dados para a biblioteca da UFAL, além de se mostrar um instrumento útil de controle periódico e facilitar o processo de produção da informação serviu de base para o estudo e aprendizado prático do autor que poderá utilizar esses conhecimentos para as etapas a serem executadas futuramente. Apesar de atingidos os objetivos propostos, é possível reconhecer que há pontos de melhoria abarcando a real necessidade da UFAL, bem como a melhoria da documentação e relação entre as tabelas porém, este trabalho não contempla toda essa complexidade em seu escopo.

A relevância deste estudo pode ser percebida no próprio motivo de sua escolha, onde reflete uma necessidade real que precisa ser implementada no ambiente acadêmico da UFAL e considerando a crescente digitalização das informações públicas e a necessidade de publicização dos dados e acesso facilitado a periódicos acadêmicos, ficando a implementação do sistema imperiosamente sugerida como trabalhos futuros uma vez que a base de dados aqui modelada irá servir como exemplo e ponto de partida para essa implementação futura.

Ao estudar e implementar a modelagem de dados relacional foi possível entender a complexidade de se projetar um sistema capaz de atender a requisitos

pré-estabelecidos, otimizando os processos e facilitando a gestão de dados. A teoria obtida por meio do levantamento bibliográfico acrescido da experiência prática, proporcionou uma vasta compreensão dos desafios reais enfrentados pelos profissionais de TI em suas práticas profissionais, em especial no desenvolvimento de bancos de dados. Com isso, observa-se que tais aprendizados contribuíram para o desenvolvimento profissional teórico e prático do acadêmico.

REFERÊNCIAS

BRMODELO. v3.31. [2025?] Disponível em:

<https://www.brmodeloweb.com/lang/pt-br/index.html>. Acesso em: 20 jun 2025.

CODD, Edgar F. **Extending the database relational model to capture more meaning**. ACM Transactions on Database Systems (TODS), v. 4, n. 4, p. 397-434, 1979.

COSTA, Alan Nicolas Da. **DESENVOLVIMENTO DE UM SISTEMA WEB PARA GESTÃO A INSTITUIÇÃO CT CRUZ AZUL DE PANAMBI**. INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FARROUPILHA CAMPUS PANAMBI (IFFAR). Disponível em: <https://arandu.iffarroupilha.edu.br/handle/itemid/259>. Acesso em: 20 jun 2025

FILL, Dorotéa Maris Estela; et. al. **Padronização da coleta de dados nas bibliotecas do SIBI/USP**. Artigos. Perspect. ciênc. inf. 11 (2). Ago 2006. Doi: <https://doi.org/10.1590/S1413-99362006000200007>. Disponível em: <https://www.scielo.br/j/pci/a/FcW46HxvZBx9HDxGnbZ7Vtf/?format=pdf&lang=pt>. Acesso em: 20 jun 2025

FOX, E. A.; MARCHIONINI, G. **Digital libraries**. Communications of the ACM, v. 44, n. 1, p. 88-89, May 2001.

HEUSER, C. A. **Projeto de banco de dados**. Porto Alegre: Grupo A, 2011. v. 4.

MASIERO, PAULO CESAR; et. al. **A Biblioteca Digital de Teses e Dissertações da Universidade de São Paulo**. Ci. Inf., Brasília, v. 30, n. 3, p. 34-41, set./dez. 2001. Artigo recebido em 19/11/2001. Disponível em:

<https://www.scielo.br/j/ci/a/8gnnDWKLyWpSywctJckDxLd/?format=pdf&lang=pt>.

Acesso em: 20 jun 2025

MICROSOFT WORKBENCH. v.8.0.42. [2025?]. Disponível em:

<https://www.mysql.com/products/workbench/>. Acesso em: 20 jun 2025

MORETTIN, P. A.; BUSSAB, W.O. **Estatística básica**. 5. ed. São Paulo: Saraiva, 2003.

MÜLLER, R. J. **Database Design for Smarties**: Using UML for Data Modeling.

Morgan Kaufmann, 2002.

MySQL. Disponível em: <https://www.mysql.com/>. Acesso em: 20 Jun. 2025

SOMMERVILLE, I. Engenharia de software. 8. ed. Pearson, 2007.

VERGUEIRO, W. C. S.; CARVALHO, T. **Indicadores de qualidade em bibliotecas universitárias brasileiras** In: CONGRESSO BRASILEIRO DE BIBLIOTECONOMIA E DOCUMENTAÇÃO, 19., 2000, Porto Alegre. Anais... Porto Alegre: Associação Riograndense de Bibliotecários, 2000. 1 CD-ROM.

XAMPP. v.8.2.4-0. [2025?]. Disponível em:

https://www.apachefriends.org/pt_br/index.html. Acesso em: 20 jun 2025