



**UNIVERSIDADE FEDERAL DE SANTA MARIA**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ELETRÔNICA E COMPUTAÇÃO**  
**CIÊNCIA DE DADOS I**

Bruno Henrique Spies  
Êmilly Farias Brunning  
Mathias Cirolini Michelotti  
Nathan Cezar  
Vinícius Gabriel Schultz

**Trabalho para a disciplina de ciência de dados I**  
Modelagem de um banco de dados relacional



**UNIVERSIDADE FEDERAL DE SANTA MARIA**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ELETRÔNICA E COMPUTAÇÃO**  
**CIÊNCIA DE DADOS I**

**Nomes do Alunos(as):**

Bruno Henrique Spies  
Êmilly Farias Brunning  
Mathias Cirolini Michelotti  
Nathan Cezar  
Vinícius Gabriel Schultz

**Matrículas:**

202220187  
2023520081  
202220937  
202121115  
202220819

**E-mail:**

email

**Curso:**

Engenharia de Computação

**Nome do Profesor(a):** Gabriel Machado Lunardi

# Sumário

<b>1</b>	<b>RESUMO .....</b>	<b>1</b>
<b>2</b>	<b>INTRODUÇÃO.....</b>	<b>2</b>
<b>3</b>	<b>METODOLOGIA.....</b>	<b>4</b>
<b>4</b>	<b>DESENVOLVIMENTO .....</b>	<b>5</b>
4.1	Modelo Conceitual .....	5
4.2	Modelo Lógico.....	10
4.3	Modelo Físico .....	12
4.3.1	<b>Projeto em SQL .....</b>	<b>12</b>
4.3.2	<b>Preenchimento do Modelo .....</b>	<b>17</b>
4.4	Interface Gráfica e Integração com o Banco de Dados .....	23
4.4.1	<b>Integração com o Banco de Dados .....</b>	<b>23</b>
4.4.1.1	<u>Conexão com o Banco de Dados .....</u>	<u>24</u>
4.4.1.2	<u>Operações de Leitura (Consultas) .....</u>	<u>24</u>
4.4.1.3	<u>Consultas Simples .....</u>	<u>26</u>
4.4.1.4	<u>Consultas com Parâmetros .....</u>	<u>26</u>
4.4.1.5	<u>Consultas Complexas .....</u>	<u>26</u>
4.4.1.6	<u>Manipulação de Dados.....</u>	<u>27</u>
4.4.1.7	<u>Limpeza de tabelas existentes .....</u>	<u>28</u>
4.4.1.8	<u>Uso de Estruturas de Interface.....</u>	<u>28</u>
4.4.1.9	<u>Tratamento de Exceções .....</u>	<u>28</u>
4.4.2	<b>Funcionamento da Interface.....</b>	<b>29</b>
4.4.2.1	<u>Cadastrar Dados nas Tabelas.....</u>	<u>29</u>
4.4.2.2	<u>Remover Dados das Tabelas.....</u>	<u>32</u>
4.4.2.3	<u>Editar Dados das Tabelas .....</u>	<u>33</u>
4.4.2.4	<u>Buscar Dados nas Tabelas.....</u>	<u>35</u>
4.4.2.5	<u>Janela de Centros .....</u>	<u>37</u>
4.4.2.6	<u>Janela de Alunos .....</u>	<u>37</u>
4.4.2.7	<u>Janela de Publicações .....</u>	<u>38</u>
4.4.2.8	<u>Janela de Bibliotecas .....</u>	<u>40</u>
4.4.2.9	<u>Janela de Empréstimos .....</u>	<u>40</u>
4.4.2.10	<u>Janela de Funcionários .....</u>	<u>41</u>
4.4.2.11	<u>Janela de Autores .....</u>	<u>43</u>
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>44</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>45</b>

# 1 Resumo

O objetivo central deste trabalho foi aplicar técnicas de modelagem de banco de dados relacionais em todas as etapas de seu desenvolvimento, desde a modelagem conceitual até a implementação física e a criação de uma aplicação prática. O projeto foi realizado em grupo, visando não apenas a consolidação dos conhecimentos técnicos, mas também o desenvolvimento de habilidades interpessoais, como comunicação, colaboração e gestão de tarefas.

O problema escolhido para a aplicação foi o gerenciamento de bibliotecas universitárias, uma demanda relevante, considerando a quantidade de informações a serem organizadas e a necessidade de acessos eficientes. Assim, o trabalho destacou a importância dos bancos de dados na tecnologia e gestão de informações, mostrando como sistemas computacionais podem assegurar a integridade e a escalabilidade dos dados, promovendo a eficiência operacional.

Na primeira etapa, foram desenvolvidos os Modelos Conceitual e Lógico do banco de dados, utilizando técnicas de modelagem aprendidas em aula. A segunda etapa focou na implementação física do modelo no SGBD MariaDB, através de comandos SQL, e na criação de uma interface gráfica funcional que permite interações como inserção, consulta e remoção de dados das tabelas. A interface gráfica não só validou o modelo físico como também proporcionou uma experiência prática de integração entre banco de dados e aplicação.

A metodologia adotada incluiu o uso de ferramentas de modelagem de dados, linguagens de programação e boas práticas de versionamento no GitHub, conforme especificado pelo professor. Além disso, cada etapa foi documentada de acordo com normas acadêmicas, incluindo a contextualização do problema, a descrição técnica do projeto e o detalhamento dos testes e resultados obtidos.

Este relatório apresenta o processo completo de construção do banco de dados e desenvolvimento da aplicação, abordando desde a concepção do problema até os resultados alcançados. Ao final, são refletidos os aprendizados obtidos e sugestões para futuros aprimoramentos, reforçando a importância da prática e integração dos conceitos para o preparo profissional dos alunos.

## 2 Introdução

Gerir uma fábrica, empresa, banco ou até mesmo pequenas lojas, em qualquer cenário, é um desafio que exige bastante atenção e organização. O expressivo número de dados que necessitam ser armazenados e gerenciados precisam ser tratados com bastante delicadeza para garantir a confiabilidade, integridade e segurança, além de mantê-los organizados afim de facilitar sua consulta e leitura.

Uma forma prática, atual e tecnológica para fazer isso é utilizando um sistema computacional, pois automatiza todos os processos e métodos, garante a integridade dos dados, os armazena e mantém o histórico de alterações e agiliza a gerência das informações e a produtividade das tarefas. A grande maioria dos estabelecimentos adotam essa abordagem para gerir todo o negócio, de modo que se tornou o padrão atualmente.

Uma peça muito importante para esse sistema é a implementação de um banco de dados para armazenar e gerenciar as informações, sendo o gerenciamento de dados o assunto principal da disciplina em questão. Para a consolidação e prática do aprendizado, foi proposto pelo docente a implementação de um banco de dados, abrangendo o modelo conceitual, o modelo lógico, o modelo físico e uma interface gráfico, de um caso real ou fictício, com o intuito de que ao final do processo todos os alunos tenham compreendido perfeitamente os assuntos discutidos em aula e estejam aptos a realizar essa tarefa em suas futuras carreiras quando lhes forem solicitada.

O modelo conceitual de um banco de dados descreve o seu comportamento de forma abstrata e independente do sistema de gerenciamento de banco de dados (SGDB). Comumente é expresso em forma de diagrama utilizando a abordagem entidade-relacionamento (ER) que é a mais difundida na comunidade de programadores.

O modelo lógico baixa um pouco o nível e é dependente do SGBD escolhido. Ele pode ser expresso por diagramas assim como o modelo conceitual por meio do chamado modelo "pé de galinha" ou então com descrições textuais e tabelas.

O modelo físico é a implementação em código do banco de dados usualmente feita em SQL. Aqui é informado os tipos dos atributos e suas regras de negócio, além de definir campos opcionais ou não, auto-incremento das chaves e entre outras informações relevantes para a aplicação final.

A interface gráfica tem o intuito de capacitar os discentes a integrar o banco de dados criado até o modelo físico a um software real, de modo que necessitem realizar inserções, edições, remoções e buscas no banco e assim aplicando o restante do conteúdo a cerca de SQL aprendido.

Para o trabalho foi escolhido realizar a informatização de um banco de dados para gerencia das bibliotecas universitárias. A turma foi separada em grupos de quatro a cinco integrantes para a realização da tarefa. Além de aplicação dos conhecimentos, o docente avaliará também aspectos qualitativos como trabalho em equipe e oratória em uma apresentação oral do trabalho para a turma.

O enunciado formulado para a realização do trabalho foi o seguinte:

Cada **biblioteca** tem um **nome**, **endereço** e seu **código** e pode **pertencer** a nenhum ou mais **centros** da faculdade e **possuir** nenhuma ou várias **publicações**.

Os **centros** têm um **nome**, **sigla** e seu **código** para identificação e podem ou não ter **bibliotecas** e **alunos** associados.

**Alunos** são cadastrados com **nome**, **endereço** e **matrícula** para os identificar e são **locados** a um **centro** e podem **requisitar** nenhum ou vários **empréstimos** nas **bibliotecas**.

Cada **empréstimo** possui **data**, **hora** e um **código** para identificá-lo, pode ser **requisitado** por somente um **aluno**, **emprestar** somente uma **publicação** e é **efetuado** por um **funcionário**.

Cada **funcionário** tem um **nome**, seu **salário** e um **código** para ser identificado e pode **efetuar** nenhum ou vários **empréstimos**.

Cada **publicação** tem **ano**, **tipo** e um **código**, deve estar de **posse** de somente uma **biblioteca**, pode ter **associada** nenhum ou vários **empréstimos**, nem sempre precisam ser **acadêmicas**, de **autoajuda** ou de **literatura** e pode ser **escrita** por um ou vários **autores**. As **publicações acadêmicas** têm **área** e **edição**; as de **autoajuda**, **assunto**; e as de **literatura**, **gênero textual**.

Cada **autor** tem um **código**, **nome** e **país** e pode **escrever** nenhuma ou várias **publicações**.

### 3 Metodologia

Inicialmente, o grupo se reuniu para decidir o tema do trabalho. Em seguida, as tarefas foram distribuídas entre os integrantes, de forma a garantir a uniformidade e a participação de todos. No final, as contribuições individuais foram reunidas para formar o trabalho completo. Todo o desenvolvimento foi realizado de forma colaborativa por meio do repositório no GitHub, garantindo a organização e o versionamento adequado dos arquivos.

Para a elaboração do Diagrama Entidade-Relacionamento (DER) e do Modelo Lógico, utilizou-se o software *brModelo*, na versão *desktop*. O relatório foi redigido em conformidade com as normas da ABNT, utilizando o editor *Overleaf* em linguagem  $\text{\LaTeX}$ . A apresentação de slides foi criada com o auxílio da plataforma *Canva*, enquanto as imagens foram editadas no *GIMP*, configurado com o pacote *PhotoGIMP*, que ajusta sua interface para se assemelhar ao Adobe Photoshop.

O modelo físico foi implementado no SGBD *MariaDB*; no entanto, para aplicações simples, utilizou-se o *SQLLiteOnline*. A interface gráfica foi desenvolvida em *Java Swing* utilizando o *NetBeans*, e, para alguns diagramas complementares, empregou-se o software online *LucidChart*.

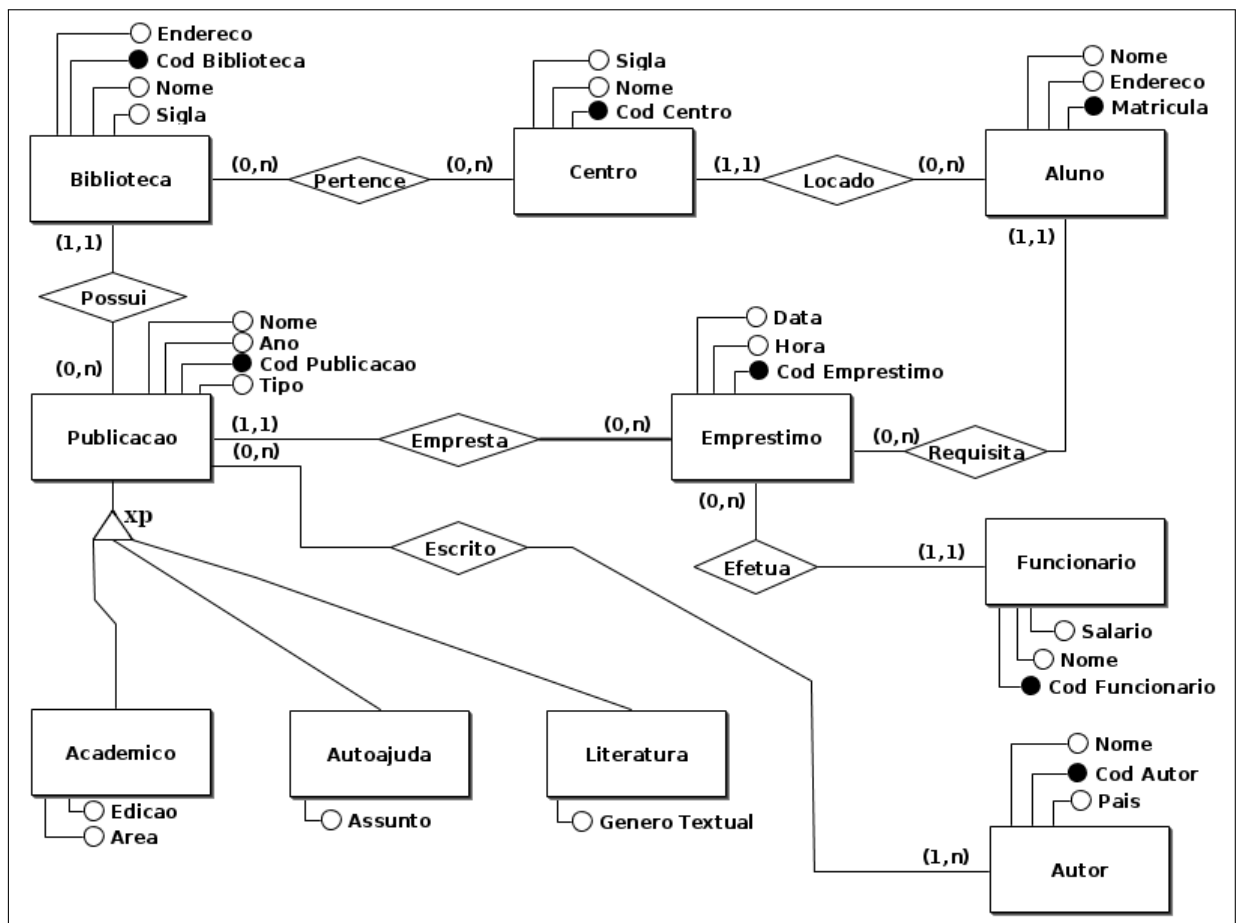
Todos os arquivos do projeto foram centralizados e compartilhados no repositório do GitHub intitulado `grupo7-CDI-20242-tg2`. O GitHub também facilitou o trabalho em equipe, permitindo o versionamento contínuo e a integração das diferentes etapas do projeto, além de documentar o progresso de forma estruturada.

## 4 Desenvolvimento

### 4.1 Modelo Conceitual

A Figura 1 abaixo se trata do Diagrama Entidade-Relacionamento desenvolvido pelo grupo, o qual representa a organização de um banco de dados para gerenciar as bibliotecas de uma universidade.

Figura 1 – Modelo Conceitual



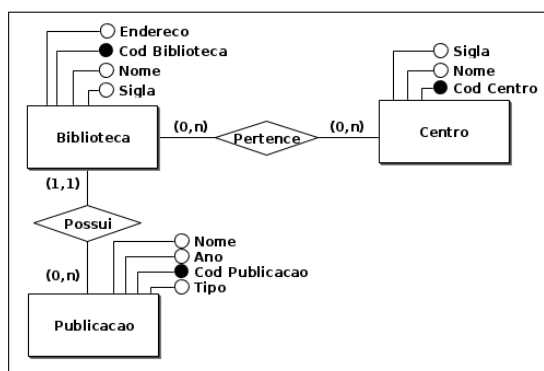
Fonte: Arquivo pessoal, 2024.

O Diagrama projetado possui as seguintes entidades: Biblioteca, Centro, além de Aluno, Publicação, Empréstimo, Funcionário e Autor, sendo que a entidade Publicação se especializa de forma parcial e exclusiva nas entidades Acadêmico, Autoajuda e Literatura.

A entidade Biblioteca possui os atributos: Cod Biblioteca (atributo identificador), Nome, Sigla e Endereço e se relaciona com as entidades Centro e Publicação. Cada biblioteca pertence a zero ou mais centros e possui zero ou mais publicações. A entidade está ilustrada na Figura 2 abaixo.



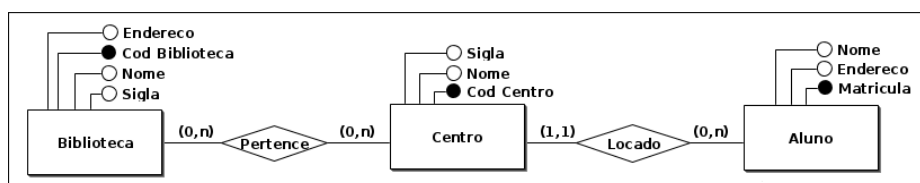
Figura 2 – Entidade Biblioteca



Fonte: Arquivo pessoal, 2024.

A entidade Centro ilustrada na Figura 52, possui os atributos: Cod\_Centro (atributo identificador), Sigla e Nome e se relaciona com as entidades Biblioteca e Aluno. Cada Centro possui zero ou mais bibliotecas e tem locado zero ou mais alunos.

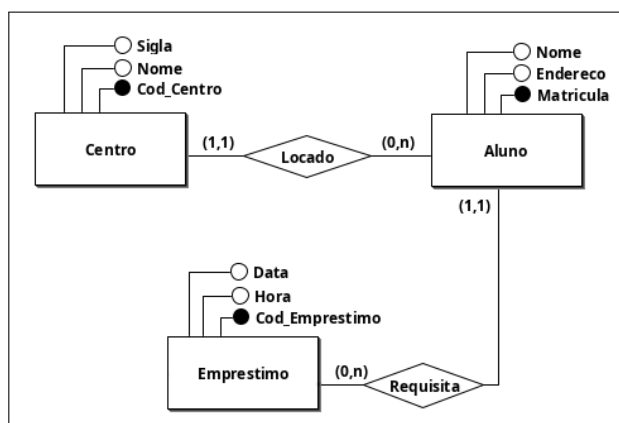
Figura 3 – Entidade Centro



Fonte: Arquivo pessoal, 2024.

A entidade Aluno ilustrada na Figura 4, possui os atributos: Matrícula (atributo identificador), Nome e Endereço e se relaciona com as entidades Centro e Empréstimo. Cada Aluno está locado a somente um centro e pode requisitar zero ou mais empréstimos de Publicações.

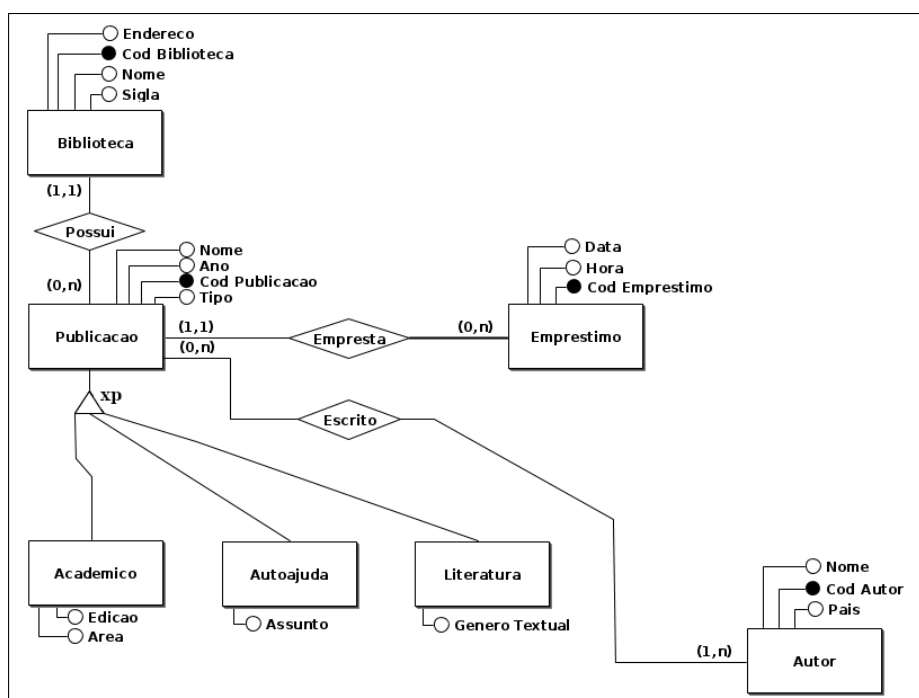
Figura 4 – Entidade Aluno



Fonte: Arquivo pessoal, 2024.

A entidade Publicação ilustrada na Figura 5, possui os seguintes atributos: Cod Publicação (atributo identificador), Ano, Nome e Tipo. O atributo Tipo foi adicionado pelo fato de que a entidade Publicação é genérica, e possui generalização parcial e exclusiva, portanto a presença do atributo Tipo é obrigatória para incluir generalizações não descritas no diagrama. A entidade se generaliza nas seguintes entidades especializadas: Acadêmico destinada para livros didáticos e semelhantes, Autoajuda para livros de aperfeiçoamento pessoal e Literatura dedicada à livros narrativos. Se relaciona com as entidades Biblioteca, Autor e Empréstimo. Cada publicação pertence a somente uma Biblioteca, é escrita por um ou mais Autores e pode ser emprestada zero ou muitas vezes.

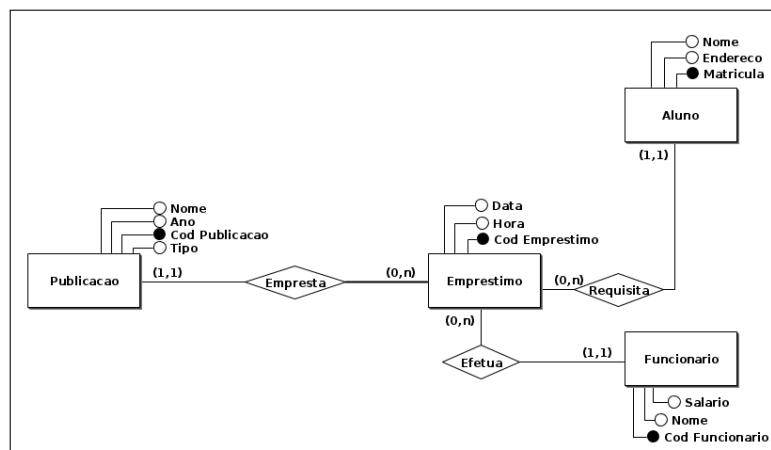
Figura 5 – Entidade Publicação



Fonte: Arquivo pessoal, 2024.

A entidade Empréstimo ilustrada na Figura 6 abaixo, é uma entidade fraca, pois depende da existência da entidade Publicação para existir e ser identificada, por isso o traçado do relacionamento desta entidade com Publicação é mais espesso. A entidade possui os seguintes atributos: Cod Empréstimo (atributo identificador), Data e Hora e se relaciona com as entidades Publicação, Funcionário e Aluno. Cada empréstimo empresta somente uma publicação, é efetuado por somente um funcionário e é requisitado por somente um aluno.

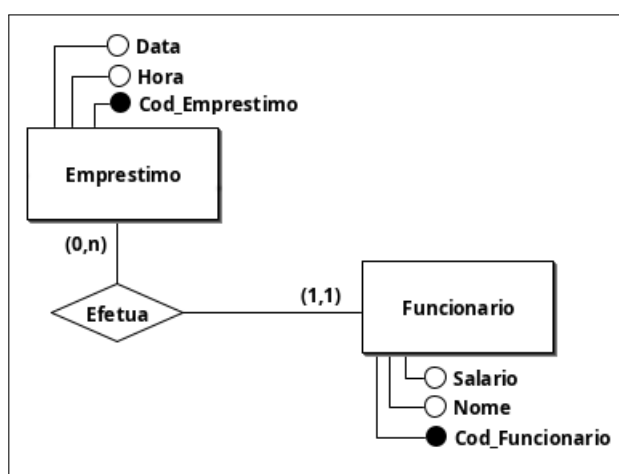
Figura 6 – Entidade Empréstimo



Fonte: Arquivo pessoal, 2024.

A entidade Funcionário possui os atributos: Cod Funcionario (atributo identificador), Salario e Nome e se relaciona com a entidade Empréstimo. Cada funcionário efetua zero ou vários empréstimos. A entidade está ilustrada na Figura 7 abaixo.

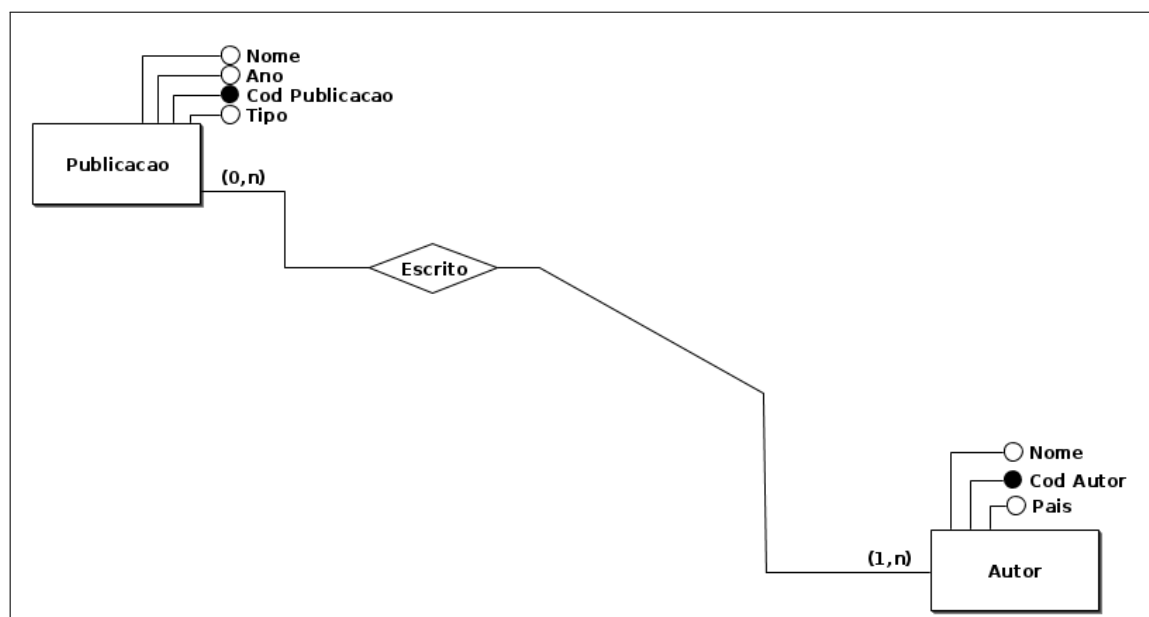
Figura 7 – Entidade Funcionário



Fonte: Arquivo pessoal, 2024.

A entidade Autor ilustrada na Figura 8 abaixo, possui os atributos: Cod Autor (atributo identificador), Nome e País (país) e se relaciona com a entidade Publicacao. Cada Autor escreveu zero ou mais publicações.

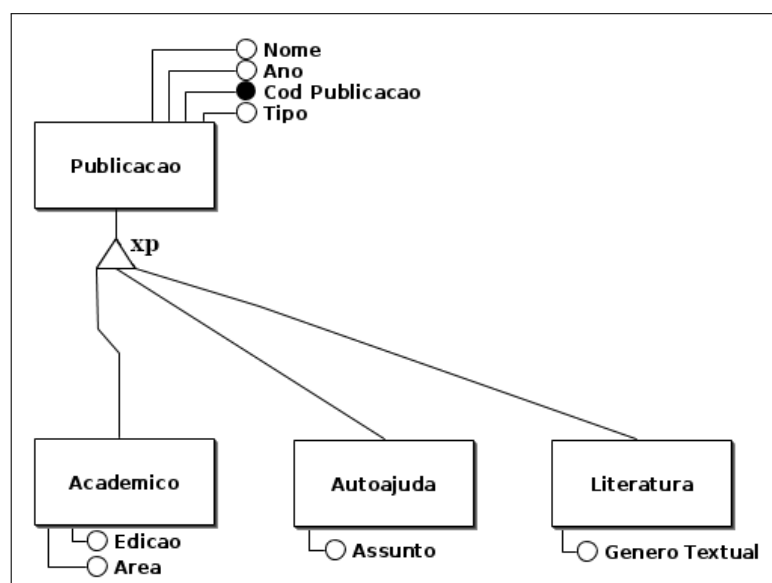
Figura 8 – Entidade Autor



Fonte: Arquivo pessoal, 2024.

A entidade Acadêmico possui os atributos específicos: Edicao e Area. A entidade Autoajuda possui o atributo específico: Assunto. A entidade Literatura possui o atributo específico: Genero Textual. Todas são generalizações da entidade Publicação e estão ilustradas na Figura 9 abaixo. As especializações são do tipo xp que indica que são exclusivas, pode ser generalizado em somente uma categoria, e parcial, pode ser generalizado em outro tipo que não esteja indicado e será armazenado no atributo Tipo da entidade Publicacao.

Figura 9 – Entidades das especializações de Publicação

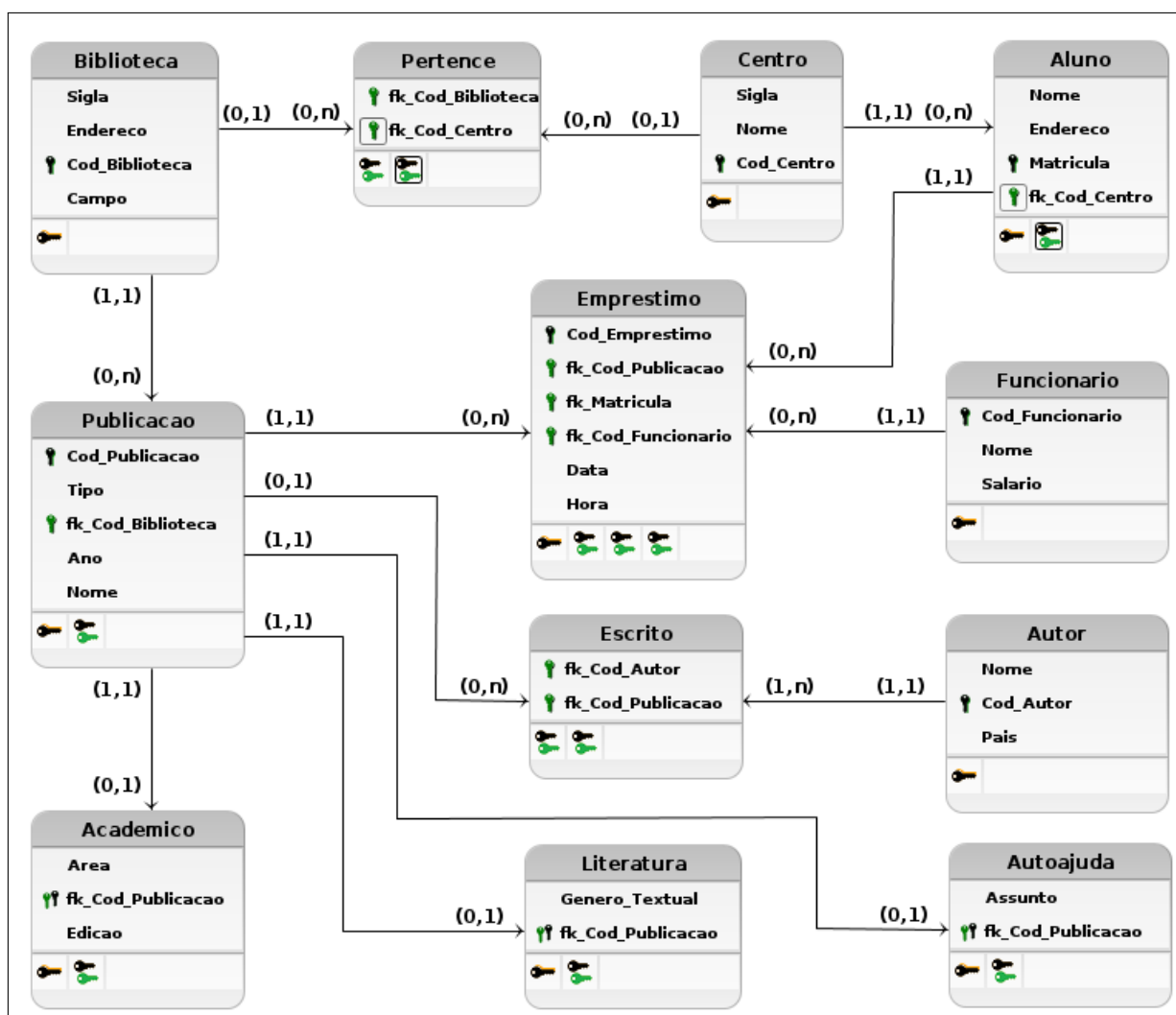


Fonte: Arquivo pessoal, 2024.

## 4.2 Modelo Lógico

Para a realização do modelo lógico, foram seguidas as regras de transformações entre modelos, sendo três regras principais de três diferentes relacionamentos que devem ser seguidas. No caso deste trabalho, apenas as regras de transformação 1:N e N:N foram necessárias. Conforme a regra N:N, *Pertence* (entre *Biblioteca* e *Centro*) virou uma tabela própria, possuindo duas chaves estrangeiras, sendo elas *fk\_Cod\_Biblioteca* e *fk\_Cod\_Centro*, usadas para demonstrar de maneira mais lógica quais atributos com identificadores únicos interagem entre si para realizar suas devidas funções. O mesmo ocorre com a criação de uma nova tabela própria chamada *Escrito*, onde pode-se perceber o surgimento de duas chaves estrangeiras, conforme pode-se analisar na Figura 10.

Figura 10 – Modelo Lógico



Fonte: Arquivo pessoal, 2024.

Além disso, foi implementando adição de colunas, entre relações  $(x,n)$  para  $(x,1)$  onde  $x$  segue a mesma lógica, podendo ser  $n$ , 0 ou 1, não tem diferença pois o que interessa no modelo lógico é apenas o último valor, com exceção nos relacionamentos 1:1, onde a cardinalidade mínima importa. Neste caso, mostra-se de extrema importância seguir as regras de transformação para relacionamentos 1:N, no qual não se cria nenhuma tabela nova, apenas adiciona-se a chave estrangeira no lado N, vale ressaltar que a chave estrangeira é sempre a chave primária de outra tabela. Essa adição de colunas funciona de maneira diferente da anterior, neste caso, a relação não se torna uma nova entidade, apenas a chave primária sendo copiada para dentro da entidade que possui  $(x,n)$  como relação. Nesse caso, *Empréstimo* recebeu a chave estrangeira de *Funcionário*, *Aluno* e *Publicação*, conforme as regras.

Como “*Publicação*” tem uma generalização, não é realizada a fusão, mesmo o relacionamento entre estes sendo  $(x,1)$  para  $(x,1)$ . Isso se deve ao fato de que, como cada tipo de publicação tem atributos diferentes que não são identificadores únicos, a fusão resultaria em uma confusão onde não teria modo de distinguir qual dos atributos são de um determinado tipo.

No caso deste trabalho, como dito anteriormente, possui-se apenas relações N:N e 1:N. As duas relações N:N foram explicadas anteriormente, entretanto, para as relações 1:N, torna-se dispensável a explicação individual de cada transformação, pois todas seguem a mesma regra. Portanto, a fim de evitar redundâncias, foi explicado a criação da tabela *Empréstimo*, cuja ideia segue-se a mesma para todos os outros relacionamentos 1:N.

### 4.3 Modelo Físico

Após a concepção do modelo lógico passou-se para o modelo físico e implementação das tabelas escritas em SQL. Foi utilizado o SGDB MariaDB pois ser o substituto do MySQL que foi adquirido pela Oracle.

#### 4.3.1 Projeto em SQL

Abaixo, na Figura 11, o código de criação da tabela Biblioteca. Foi criado com uma chave primária chamada Cod\_biblioteca como inteiro e os atributos Endereco e Nome como array de caracteres, denotado por "VARCHAR", ambos como 100 caracteres, além da Sigla como VARCHAR de 10 caracteres.

Figura 11 – Criação da tabela "Biblioteca" em SQL.

```
CREATE TABLE Biblioteca (  
  Cod_Biblioteca INT PRIMARY KEY AUTO_INCREMENT,  
  Endereco VARCHAR(100) NOT NULL,  
  Nome VARCHAR(100) NOT NULL,  
  Sigla VARCHAR(10) NOT NULL  
);
```

Fonte: Arquivo pessoal, 2024.

A tabela Centro foi criada como indicado na Figura 12 abaixo. Foi criada a chave primária Cod\_Centro como INT e os atributos Sigla como um VARCHAR de 10 caracteres e UNIQUE, pois também pode ser utilizado como identificador, e Nome como um VARCHAR de 50 caracteres.

Figura 12 – Criação da tabela "Centro" em SQL.

```
CREATE TABLE Centro (  
  Cod_Centro INT PRIMARY KEY AUTO_INCREMENT,  
  Sigla VARCHAR(10) UNIQUE NOT NULL,  
  Nome VARCHAR(50) NOT NULL  
);
```

Fonte: Arquivo pessoal, 2024.

Como o relacionamento entre as entidades Biblioteca e Centro é N:N isso infere na criação de uma nova tabela chamada Pertence. A criação dessa tabela está informada na Figura 13 abaixo. Ela contém somente a chave primária formada pela composição das chaves estrangeiras fk\_Cod\_Biblioteca, que referencia Biblioteca, e fk\_Cod\_Centro, que referencia Centro.

Figura 13 – Criação da tabela "Pertence"em SQL.

```
CREATE TABLE Pertence (
    fk_Cod_Biblioteca INT NOT NULL,
    fk_Cod_Centro INT NOT NULL,
    FOREIGN KEY (fk_Cod_Biblioteca) REFERENCES Biblioteca(Cod_Biblioteca),
    FOREIGN KEY (fk_Cod_Centro) REFERENCES Centro(Cod_Centro)
);
```

Fonte: Arquivo pessoal, 2024.

Na Figura 14 abaixo está informada a criação da tabela Aluno. Tem uma chave primária chamada Matricula, criada como INT, e os atributos Endereco e Nome como VARCHAR de 100 e 50 caracteres respectivamente, além de uma chave estrangeira chamada fk\_Cod\_Centro criada como INT, que referencia Centro para implementar o relacionamento 1:N que ocorre por adição de coluna de Centro em Aluno.

Figura 14 – Criação da tabela "Aluno"em SQL.

```
CREATE TABLE Aluno (
    Matricula INT PRIMARY KEY AUTO_INCREMENT,
    Endereco VARCHAR(100) NOT NULL,
    Nome VARCHAR(50) NOT NULL,
    fk_Cod_Centro INT NOT NULL,
    FOREIGN KEY (fk_Cod_Centro) REFERENCES Centro(Cod_centro)
);
```

Fonte: Arquivo pessoal, 2024.

A criação da tabela Funcionario está indicada na Figura 15 abaixo. Conta com uma chave primária chamada Cod\_Funcionario como INT, e com os atributos Salario como DECIMAL de 10 dígitos, sendo 2 para as casas decimais, e Nome como VARCHAR de 50 caracteres.

Figura 15 – Criação da tabela "Funcionario"em SQL.

```
CREATE TABLE Funcionario (
    Cod_Funcionario INT PRIMARY KEY AUTO_INCREMENT,
    Salario DECIMAL(10,2) NOT NULL,
    Nome VARCHAR(50) NOT NULL
);
```

Fonte: Arquivo pessoal, 2024.

A Figura 16 abaixo indica a criação da tabela Publicacao. Conta com a chave primária Cod\_Publicacao como INT, o atributo opcional Tipo como VARCHAR de 50 caracteres e os atributos Ano como DATE e Nome como VARCHAR de 100 caracteres, além da chave estrangeira fk\_Cod\_Biblioteca, criada como INT que referencia Biblioteca para implementar o relacionamento 1:N entre Publicacao e Biblioteca por adição de coluna.



Figura 16 – Criação da tabela "Publicacao"em SQL.

```
CREATE TABLE Publicacao (
  Cod_Publicacao INT PRIMARY KEY AUTO_INCREMENT,
  Tipo VARCHAR(50) NULL,
  Ano YEAR NOT NULL,
  Nome VARCHAR(100) NOT NULL,
  fk_Cod_Biblioteca INT NOT NULL,
  FOREIGN KEY (fk_Cod_Biblioteca) REFERENCES Biblioteca(Cod_Biblioteca)
);
```

Fonte: Arquivo pessoal, 2024.

O atributo Tipo é opcional utilizando a palavra reservada NULL pois a generalização/especialização é parcial, portanto esse atributo é utilizado caso não se especialize nos tipos pré-definidos. Para as especializações foram implementadas tabelas próprias para cada uma. Abaixo na Figura 17 está indicada a criação da tabela Academico, que conta com os atributos Edicao como INT e Area com VARCHAR de 50 caracteres, além da chave estrangeira fk\_Cod\_Publicacao que referencia Publicacao.

Figura 17 – Criação da tabela "Academico"em SQL.

```
CREATE TABLE Academico (
  Edicao INT NOT NULL,
  Area VARCHAR(50) NOT NULL,
  fk_Cod_Publicacao INT NOT NULL,
  FOREIGN KEY (fk_Cod_Publicacao) REFERENCES Publicacao(Cod_Publicacao)
);
```

Fonte: Arquivo pessoal, 2024.

Além da especialização Academico também há a Literatura que sua criação está indicada na Figura 18 abaixo. Ela conta com o atributo Genero\_Textual que é um VARCHAR de 50 caracteres além da chave estrangeira fk\_Cod\_Publicacao que referencia Publicacao.

Figura 18 – Criação da tabela "Literatura"em SQL.

```
CREATE TABLE Literatura (
  Genero_Textual VARCHAR(50) NOT NULL,
  fk_Cod_Publicacao INT NOT NULL,
  FOREIGN KEY (fk_Cod_Publicacao) REFERENCES Publicacao(Cod_Publicacao)
);
```

Fonte: Arquivo pessoal, 2024.

Ainda há a especialização Autoajuda que tem sua criação indicada na Figura 19 abaixo. Ela conta com o atributo Assunto como VARCHAR de 50 caracteres e a chave estrangeira fk\_Cod\_Publicacao que referencia Publicacao.

Figura 19 – Criação da tabela "Autoajuda"em SQL.

```
CREATE TABLE Autoajuda (
  Assunto VARCHAR(50) NOT NULL,
  fk_Cod_Publicacao INT NOT NULL,
  FOREIGN KEY (fk_Cod_Publicacao) REFERENCES Publicacao(Cod_Publicacao)
);
```

Fonte: Arquivo pessoal, 2024.

Abaixo, na Figura 20 está indicada a criação da tabela Autor. Tem uma chave primária nomeada Cod\_Autor como INT e os atributos Nome e Pais, ambos como VARCHAR de 50 caracteres.

Figura 20 – Criação da tabela "Autor"em SQL.

```
CREATE TABLE Autor (
  Cod_Autor INT PRIMARY KEY AUTO_INCREMENT,
  Nome VARCHAR(50) NOT NULL,
  Pais VARCHAR(50) NOT NULL
);
```

Fonte: Arquivo pessoal, 2024.

O relacionamento entre Publicacao e Autor também é N:N o que obriga a criação de uma tabela ptópria para sua implementação. A criação da tabela Escrito está indicada abaixo na Figura 21. Ela tem somente as chaves estrangeiras das duas entidades; fk\_Cod\_Publicacao que referencia Publicacao e fk\_Cod\_Autor que referencia Autor, ambas INT.

Figura 21 – Criação da tabela "Escrito"em SQL.

```
CREATE TABLE Escrito (
  fk_Cod_Publicacao INT NOT NULL,
  fk_Cod_Autor INT NOT NULL,
  FOREIGN KEY (fk_Cod_Publicacao) REFERENCES Publicacao(Cod_Publicacao),
  FOREIGN KEY (fk_Cod_Autor) REFERENCES Autor(Cod_Autor)
);
```

Fonte: Arquivo pessoal, 2024.

Por fim, na Figura 22 abaixo o código de criação da tabela Emprestimo. Ela conta com a chave primária Cod\_Emprestimo, criada como INT, os atributos Data\_, criado como DATE e com o "\_" para diferenciá-lo da palavra reservada DATA, Hora criada como TIME e também as chaves estrangeiras fk\_Cod\_Publicacao que referencia Publicacao, fk\_Matricula que referencia Aluno e fk\_Cod\_Funcionario que referencia Funcionario, todos criados como INT com o intuito de implementar por adição de coluna os relacionamentos 1:N com as entidades Publicacao, Aluno e Funcionario.

Figura 22 – Criação da tabela "Emprestimo" em SQL.

```
CREATE TABLE Emprestimo (
  Cod_Emprestimo INT PRIMARY KEY AUTO_INCREMENT,
  Data_ DATE NOT NULL,
  Hora TIME NOT NULL,
  fk_Cod_Publicacao INT NOT NULL,
  fk_Matricula int NOT NULL,
  fk_Cod_Funcionario INT NOT NULL,
  FOREIGN KEY (fk_Cod_Publicacao) REFERENCES Publicacao(Cod_Publicacao),
  FOREIGN KEY (fk_Matricula) REFERENCES Aluno(Matricula),
  FOREIGN KEY (fk_Cod_Funcionario) REFERENCES Funcionario(Cod_Funcionario)
);
```

Fonte: Arquivo pessoal, 2024.

Observa-se que todas as chaves primárias foram criadas com AUTO\_INCREMENT para que o usuário não precise se preocupar com os valores das chaves e que todos os atributos obrigatórios e chaves estrangeira foram criados com NOT NULL, sendo o único opcional o Tipo da entidade Publicacao por ser parte da generalização/especialização.

Com isso, todas as tabelas do modelo lógico indicado na Figura 10 foram implementadas em SQL. Outras aplicações em SQL, como inserções, remoções e busca, serão abordadas na Seção 4.4 referente à criação da interface gráfica que foi onde surgiram as necessidades de implementar essas aplicações.

### 4.3.2 Preenchimento do Modelo

Após a criação do modelo físico utilizando os códigos acima, as tabelas foram populadas utilizando códigos como os do exemplo da Figura 23.

Figura 23 – Exemplo de Código SQL para popular as tabelas

```
INSERT INTO Centro (Cod_Centro, Sigla, Nome)
VALUES
(01, 'CT', 'Centro de Tecnologia'),
(02, 'CCR', 'Centro de Ciências Rurais'),
(03, 'CCSH', 'Centro de Ciências Sociais e Humanas'),
(04, 'CAL', 'Centro de Artes e Letras'),
(05, 'CCNE', 'Centro de Ciências Naturais e Exatas'),
(06, 'CTISM', 'Colégio Técnico Industrial de Santa Maria'),
(07, 'CEFD', 'Centro de Educação Física e Desporto'),
(08, 'CE', 'Centro de Educação'),
(09, 'POLI', 'Colégio Politécnico'),
(10, 'CCS', 'Centro de Ciências da Saúde');
```

Fonte: Arquivo pessoal, 2024.

A Figura 24 contém uma ilustração das tabelas após terem sido adicionadas ao banco de dados com o MariaDB.

Figura 24 – Tabelas da Aplicação de Gerenciamento de Bibliotecas

```
MariaDB [cd_ii_work]> show tables;
+-----+
| Tables_in_cd_ii_work |
+-----+
| academico             |
| aluno                |
| autoajuda             |
| autor                 |
| biblioteca            |
| centro                |
| emprestimo            |
| escrito               |
| funcionario           |
| literatura            |
| pertence              |
| publicacao            |
+-----+
```

Fonte: Arquivo pessoal, 2024.

As figuras a seguir ilustrarão os dados que foram populados nas tabelas, e que posteriormente estarão disponíveis para consulta, visualização e edição na interface gráfica.

Figura 25 – Tabelas Aluno no MariaDB

```
MariaDB [cd_ii_work]> select * from aluno;
```

Matricula	Endereco	Nome	fk_Cod_Centro
202220000	Avenida Augusta	José Mourinho	1
202220001	Avenida Paulista	Carlos Ancelotti	2
202220002	Rua das Flores	Pep Guardiola	3
202220003	Rua do Sol	Roger Machado	4
202220004	Avenida Brasil	Zinedine Zidane	5
202220005	Rua da Liberdade	Diego Simeone	3
202220006	Rua do Porto	Renato Gaúcho	7
202220007	Rua Rio de Janeiro	Antonio Conte	8
202220008	Avenida São João	Arsène Wenger	3
202220009	Rua do Carmo	Frank Lampard	10
202220010	Rua das Palmeiras	David Moyes	6
202220011	Avenida dos Eucaliptos	Claudio Ranieri	2
202220012	Rua da Paz	Marcelo Bielsa	3
202220013	Rua do Parque	Tite	4
202220014	Avenida do Mar	Luis Enrique	5

Fonte: Arquivo pessoal, 2024.

Figura 26 – Tabelas Publicação no MariaDB

```
MariaDB [cd_ii_work]> select * from publicacao;
```

Cod_Publicacao	Tipo	Ano	Nome	fk_Cod_Biblioteca
1	Acadêmico	2020	Introdução à Ciência de Dados	1
2	Acadêmico	2019	Matemática Avançada	2
3	Acadêmico	2018	Fundamentos de Engenharia	3
4	Literatura	2021	Romance Clássico	4
5	Autoajuda	2022	O Poder do Hábito	5
6	Literatura	2017	O Senhor dos Anéis	6
7	Acadêmico	2020	Física para Engenheiros	7

Fonte: Arquivo pessoal, 2024.

Figura 27 – Tabelas Academico no MariaDB

```
MariaDB [cd_ii_work]> select * from academico;
```

Edicao	Area	fk_Cod_Publicacao
1	Ciência de Dados	1
2	Matemática	2
3	Engenharia Elétrica	3
4	Romance	4
5	Psicologia	5
6	Ficção Fantástica	6
7	Física	7

Fonte: Arquivo pessoal, 2024.

Figura 28 – Tabelas Autoajuda no MariaDB

```
MariaDB [cd_ii_work]> select * from autoajuda;
```

Assunto	fk_Cod_Publicacao
Desenvolvimento Pessoal	5

Fonte: Arquivo pessoal, 2024.

Figura 29 – Tabelas Literatura no MariaDB

```
MariaDB [cd_ii_work]> select * from literatura;
```

Genero_Textual	fk_Cod_Publicacao
Ficção Científica	4
Romance	6
Aventura	7

Fonte: Arquivo pessoal, 2024.

Figura 30 – Tabelas Aluno no MariaDB

```
MariaDB [cd_ii_work]> select * from aluno;
```

Matricula	Endereco	Nome	fk_Cod_Centro
202220000	Avenida Augusta	José Mourinho	1
202220001	Avenida Paulista	Carlos Ancelotti	2
202220002	Rua das Flores	Pep Guardiola	3
202220003	Rua do Sol	Roger Machado	4
202220004	Avenida Brasil	Zinedine Zidane	5
202220005	Rua da Liberdade	Diego Simeone	3
202220006	Rua do Porto	Renato Gaúcho	7
202220007	Rua Rio de Janeiro	Antonio Conte	8
202220008	Avenida São João	Arsène Wenger	3
202220009	Rua do Carmo	Frank Lampard	10
202220010	Rua das Palmeiras	David Moyes	6
202220011	Avenida dos Eucaliptos	Claudio Ranieri	2
202220012	Rua da Paz	Marcelo Bielsa	3
202220013	Rua do Parque	Tite	4
202220014	Avenida do Mar	Luis Enrique	5

Fonte: Arquivo pessoal, 2024.

Figura 31 – Tabelas Autor no MariaDB

```
MariaDB [cd_ii_work]> select * from autor;
```

Cod_Autor	Nome	Pais
1	J.K. Rowling	Reino Unido
2	George Orwell	Reino Unido
3	Clarice Lispector	Brasil
4	Gabriel García Márquez	Colômbia
5	Leo Tolstoy	Rússia
6	Carlos Drummond	Brasil
7	Jane Austen	Reino Unido
8	Mark Twain	Estados Unidos
9	Machado de Assis	Brasil
10	Ernest Hemingway	Estados Unidos

Fonte: Arquivo pessoal, 2024.

Figura 32 – Tabelas Biblioteca no MariaDB

```
MariaDB [cd_ii_work]> select * from biblioteca;
```

Cod_Biblioteca	Endereco	Nome	Sigla
1	Street 1	Biblioteca Central	BC
2	Street 2	Biblioteca Setorial do Centro de Tecnologia	BSCT
3	Street 3	Biblioteca Setorial do Centro de Ciências Rurais	BSCCR
4	Street 4	Biblioteca Setorial do Centro de Ciências Sociais e Humanas	BSCCSH
5	Street 5	Biblioteca Setorial do Centro de Artes e Letras	BSCAL
6	Street 6	Biblioteca Setorial do Centro de Ciências Naturais e Exatas	BSCCNE
7	Street 7	Biblioteca Setorial do Centro do Colégio Técnico Industrial de Santa Maria	BSCTISM
8	Street 8	Biblioteca Setorial do Centro de Educação Física e Desporto	BSCEFD
9	Street 9	Biblioteca Setorial do Centro de Educação	BSCE
10	Street 10	Biblioteca Setorial do Centro do Colégio Politécnico	BSPOLI
11	Street 11	Biblioteca Setorial do Centro de Ciências da Saúde	BSCCS

Fonte: Arquivo pessoal, 2024.

Figura 33 – Tabelas Centro no MariaDB

```
MariaDB [cd_ii_work]> select * from centro;
```

Cod_Centro	Sigla	Nome
1	CT	Centro de Tecnologia
2	CCR	Centro de Ciências Rurais
3	CCSH	Centro de Ciências Sociais e Humanas
4	CAL	Centro de Artes e Letras
5	CCNE	Centro de Ciências Naturais e Exatas
6	CTISM	Colégio Técnico Industrial de Santa Maria
7	CEFD	Centro de Educação Física e Desporto
8	CE	Centro de Educação
9	POLI	Colégio Politécnico
10	CCS	Centro de Ciências da Saúde

Fonte: Arquivo pessoal, 2024.

Figura 34 – Tabelas Empréstimo no MariaDB

```
MariaDB [cd_ii_work]> select * from emprestimo;
```

Cod_Emprestimo	Data_	Hora	fk_Cod_Publicacao	fk_Matricula	fk_Cod_Funcionario
1	2024-01-12	14:30:00	1	202220000	1
2	2024-01-13	10:00:00	2	202220001	2
3	2024-01-14	16:15:00	3	202220002	3
4	2024-02-20	11:20:00	4	202220003	4
5	2024-02-21	14:45:00	5	202220004	5
6	2024-03-05	09:30:00	6	202220005	6
7	2024-03-10	13:00:00	7	202220006	7

Fonte: Arquivo pessoal, 2024.



Figura 35 – Tabelas Funcionário no MariaDB

```
MariaDB [cd_ii_work]> select * from funcionario;
```

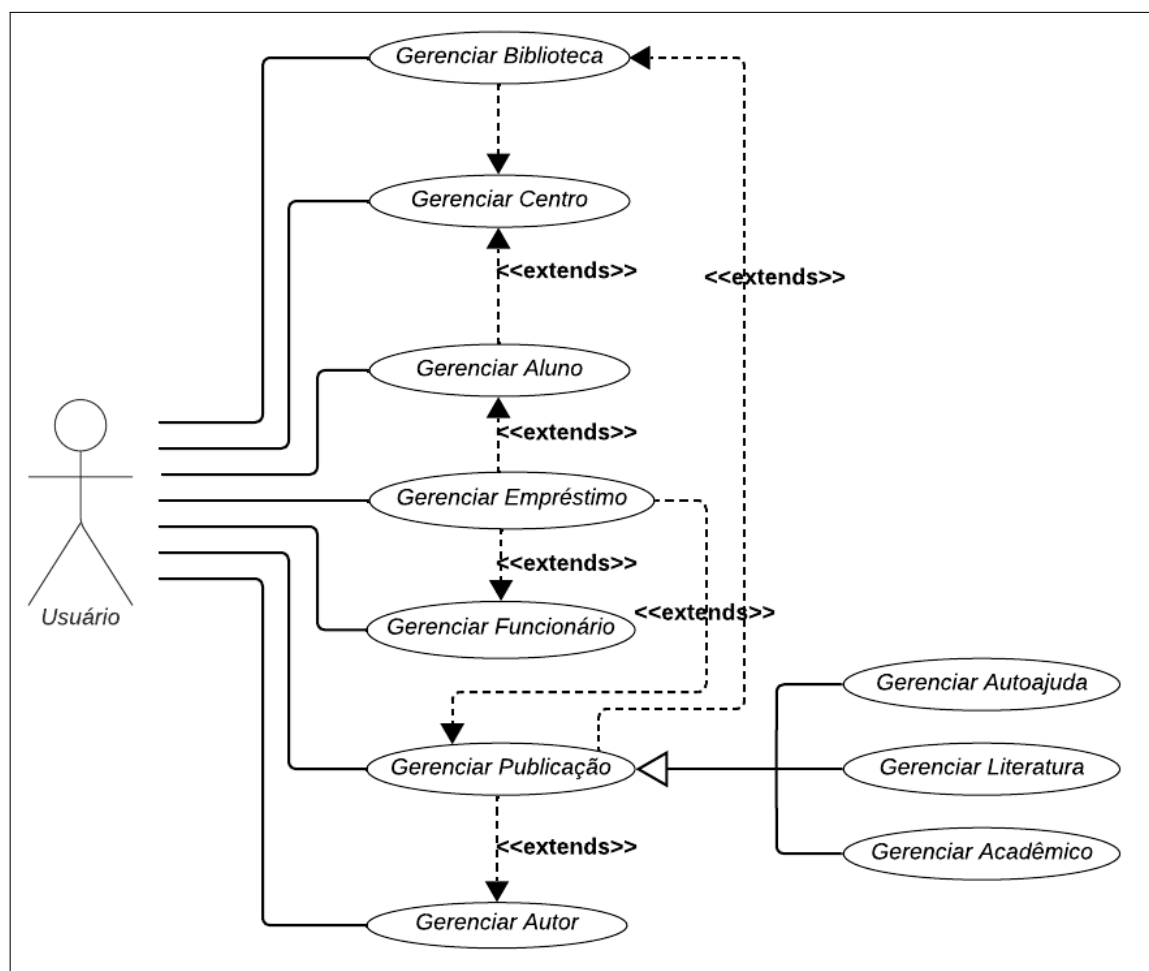
Cod_Funcionario	Salario	Nome
1	9398.98	Lionel Messi
2	10000.00	Cristiano Ronaldo
3	9500.50	Neymar Jr.
4	8700.75	Kylian Mbappé
5	11000.00	Luka Modrić
6	9900.20	Sergio Ramos
7	9600.35	Zlatan Ibrahimović
8	9200.60	Kevin De Bruyne
9	9400.80	Mohamed Salah
10	10500.90	Robert Lewandowski
11	9800.40	Virgil van Dijk
12	9150.30	Harry Kane
13	8800.45	Alan Patrick
14	9250.00	Erling Haaland
15	10050.60	Enner Valência

Fonte: Arquivo pessoal, 2024.

#### 4.4 Interface Gráfica e Integração com o Banco de Dados

Para auxiliar na criação da interface gráfica para integrar com o banco de dados foi desenvolvido o diagrama de casos de uso, indicado na Figura 36 abaixo, baseado no diagrama lógico do banco de dados.

Figura 36 – Diagrama de casos de uso da interface gráfica.



Fonte: Arquivo pessoal, 2024.

Para a criação da interface, optou-se pela utilização da linguagem de programação Java em razão do conhecimento prévio de 80% do grupo em decorrência do uso em outras disciplinas. A IDE utilizada foi o Apache NetBeans por facilitar a criação da interface em Java Swing.

##### 4.4.1 Integração com o Banco de Dados

Para integrar a interface em Java com o banco de dados em MariaDB, foi utilizada a API JDBC (Java Database Connectivity) para gerenciar operações relacionadas às tabelas da aplicação. Abaixo, estão descritos os principais aspectos dessa integração:

#### 4.4.1.1 Conexão com o Banco de Dados

Métodos conectar e fecharConexao: São responsáveis por estabelecer e encerrar a conexão com o banco de dados, garantindo que as operações sejam realizadas de forma segura e eficiente. conectar: Utiliza `SQL_connection.getConnection()` para estabelecer a conexão. fecharConexao: Fecha a conexão, caso esteja aberta.

Figura 37 – Trecho do Código de Conexão

```
public class main extends javax.swing.JFrame {  
  
    // Variável de conexão  
    private Connection conn;  
  
    // Método para conectar ao banco de dados (se necessário)  
    private void conectar() throws SQLException {  
        // Verifica se a conexão não foi estabelecida ou está fechada  
        if (conn == null || conn.isClosed()) {  
            conn = SQL_connection.getConnection(); // Estabelece a conexão com o banco  
        }  
    }  
  
    public void fecharConexao() throws SQLException {  
        if (conn != null && !conn.isClosed()) {  
            conn.close(); //  
        }  
    }  
}
```

Fonte: Arquivo pessoal, 2024.

#### 4.4.1.2 Operações de Leitura (Consultas)

O código contém diversos métodos para realizar consultas em tabelas específicas, como Centro, Aluno, Biblioteca e Empréstimo. Essas operações utilizam *Prepa-*

*redStatement* ou *Statement*, dependendo do cenário, para buscar e manipular dados.

Figura 38 – Trecho do Código Funções para Consultas

```
private void carregarPublicacoes() {
    String query = "SELECT p.Cod_Publicacao, p.Tipo, p.Ano, p.Nome, b.Sigla AS Biblioteca, " +
        "a.Edicao, a.Area, " +
        "l.Genero_Textual, " +
        "au.Assunto, " +
        "e.fk_Cod_Autor " +
        "FROM Publicacao p " +
        "JOIN Biblioteca b ON p.fk_Cod_Biblioteca = b.Cod_Biblioteca " +
        "LEFT JOIN Academico a ON p.Cod_Publicacao = a.fk_Cod_Publicacao " +
        "LEFT JOIN Literatura l ON p.Cod_Publicacao = l.fk_Cod_Publicacao " +
        "LEFT JOIN Autoajuda au ON p.Cod_Publicacao = au.fk_Cod_Publicacao " +
        "LEFT JOIN Escrito e ON p.Cod_Publicacao = e.fk_Cod_Publicacao";

    Connection conn = null;
    Statement stmt = null;
    ResultSet rs = null;

    try {
        conn = SQL_connection.getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery(query);

        DefaultTableModel model = (DefaultTableModel) TB_Publicacao.getModel();
        model.setRowCount(0);

        while (rs.next()) {
            Object[] row = {
                rs.getInt("Cod_Publicacao"), // Esta é uma coluna do tipo INT
                rs.getString("Nome"),         // Esta é uma coluna do tipo String
                rs.getInt("Ano"),             // Se "Ano" for uma coluna DATE, use rs.getDate() ao invés de rs.getInt()
                rs.getString("Biblioteca"),   // Esta é uma coluna do tipo String
                rs.getString("Tipo"),         // Esta é uma coluna do tipo String
                rs.getInt("Edicao"),           // Esta é uma coluna do tipo INT
                rs.getString("Area"),         // Esta é uma coluna do tipo String
                rs.getString("Genero_Textual"), // Esta é uma coluna do tipo String
                rs.getString("Assunto"),      // Esta é uma coluna do tipo String
            };
            model.addRow(row); // Adiciona a linha à tabela
            Itens_Empres_Publicacao.addItem(rs.getString("Nome"));
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Erro ao carregar dados: " + ex.getMessage());
    }
}
```

Fonte: Arquivo pessoal, 2024.

#### 4.4.1.3 Consultas Simples

Exemplo: `carregarCentros` e `carregarBibliotecas` realizam consultas para listar todas as entradas das respectivas tabelas.

Figura 39 – Trecho do Código Exemplo de Consulta Simples

```
// Método para preencher a tabela com os centros com o nome indicado
private void carregarCentrosNome(String nome) {
    try {
        if (conn == null || conn.isClosed()) {
            conectar();
        }

        String query = "SELECT * FROM Centro WHERE Nome=?";
        PreparedStatement ps = null;
        ps = SQL_connection.getConnection().prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
        ps.setString(1, nome);
        //Statement stmt = conn.createStatement();
        //ResultSet rs = stmt.executeQuery(query);
        ResultSet rs = ps.executeQuery();

        DefaultTableModel model = (DefaultTableModel) TB_Centro.getModel();
        model.setRowCount(0); // Limpa a tabela antes de adicionar novos dados

        while (rs.next()) {
            Object[] row = { rs.getInt("cod_Centro"), rs.getString("sigla"), rs.getString("nome") };
            model.addRow(row);
        }
        rs.close();
        ps.close();
        //stmt.close();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Erro ao carregar dados: " + ex.getMessage());
    }
}
```

Fonte: Arquivo pessoal, 2024.

#### 4.4.1.4 Consultas com Parâmetros

Métodos como `carregarCentrosNome` e `carregarCentrosSigla` permitem consultas filtradas com base em critérios específicos (exemplo: nome ou sigla). Utilização de `PreparedStatement` evita ataques de SQL Injection ao parametrizar as entradas.

#### 4.4.1.5 Consultas Complexas

Métodos como `carregarPublicacoes` e `carregarEmprestimos` realizam consultas que envolvem múltiplas tabelas e relacionamentos, utilizando JOINS para consoli-

dar os dados em uma única operação.

Figura 40 – Trecho do Código Exemplo de Consulta com Parâmetros

```
private void carregarEmprestimos() {
    // Consulta SQL para buscar os dados dos empréstimos
    String query = "SELECT e.Cod_Emprestimo, e.Data_, e.Hora, " +
        "p.Nome AS Publicacao, a.Nome AS Aluno, f.Nome AS Funcionario " +
        "FROM Emprestimo e " +
        "JOIN Publicacao p ON e.fk_Cod_Publicacao = p.Cod_Publicacao " +
        "JOIN Aluno a ON e.fk_Matricula = a.Matricula " +
        "JOIN Funcionario f ON e.fk_Cod_Funcionario = f.Cod_Funcionario";

    Connection conn = null;
    Statement stmt = null;
    ResultSet rs = null;

    try {
        conn = SQL_connection.getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery(query);

        DefaultTableModel model = (DefaultTableModel) TB_Emprestimo.getModel();
        model.setRowCount(0);

        while (rs.next()) {
            Object[] row = {
                rs.getInt("Cod_Emprestimo"),
                rs.getDate("Data_"),
                rs.getTime("Hora"),
                rs.getString("Funcionario"),
                rs.getString("Publicacao"),
                rs.getString("Aluno")
            };
            model.addRow(row);
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Erro ao carregar dados: " + ex.getMessage());
    }
}
```

Fonte: Arquivo pessoal, 2024.

#### 4.4.1.6 Manipulação de Dados

Os resultados das consultas são processados por meio de objetos `ResultSet` e transferidos para tabelas visuais (`JTable`) utilizando o modelo `DefaultTableModel`. Esse processo inclui:

#### 4.4.1.7 Limpeza de tabelas existentes

Preenchimento dinâmico das tabelas com os resultados das consultas.

Figura 41 – Trecho do Código Exemplo de Preenchimento dinâmico para Carregar Sigla do Centro

```
while (rs.next()) {
    Object[] row = { rs.getInt("cod_Centro"), rs.getString("sigla"), rs.getString("nome") };
    model.addRow(row);
}
rs.close();
ps.close();
```

Fonte: Arquivo pessoal, 2024.

#### 4.4.1.8 Uso de Estruturas de Interface

Algumas consultas, como carregarCentros e carregarAlunos, também atualizam componentes gráficos como JComboBox (Itens\_Aluno\_Centro, Itens\_Bib\_Centro), facilitando a interação do usuário.

Figura 42 – Trecho do Código Exemplo de Uso de Estruturas de Interface

```
// Método para preencher a tabela com os alunos
private void carregarAlunos() {
    try {
        if (conn == null || conn.isClosed()) {
            conectar();
        }

        // Query SQL para buscar os dados dos alunos
        //String query = "SELECT * FROM Aluno";
        String query = "SELECT Aluno.matricula, Aluno.nome, Centro.sigla, Aluno.endereco FROM Aluno JOIN Centro ON Aluno.fk_cod_centro = Centro.cod_centro";
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(query);

        DefaultTableModel model = (DefaultTableModel) TB_Aluno.getModel();
        model.setRowCount(0);

        while (rs.next()) {
            Object[] row = {
                rs.getInt("Matricula"),
                rs.getString("Nome"),
                rs.getString("Sigla"),
                rs.getString("Endereco")
            };
            model.addRow(row);
            Itens_Empres_Aluno.addItem(rs.getString("Nome"));
        }

        rs.close();
        stmt.close();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Erro ao carregar dados dos alunos: " + ex.getMessage());
    }
}
```

Fonte: Arquivo pessoal, 2024.

#### 4.4.1.9 Tratamento de Exceções

O tratamento de erros é realizado com blocos try-catch, exibindo mensagens informativas através de JOptionPane em caso de falhas na conexão ou execução das

consultas.

Figura 43 – Trecho do Código Exemplo de Tratamento de Exceções

```

    Itens_Aluno_Centro.addItem(sigla);
    Itens_Bib_Centro.addItem(sigla);

    TF_Centro_Nome.setText("");
    TF_Centro_Sigla.setText("");

    JOptionPane.showMessageDialog(null, "Centro cadastrado com sucesso!",
                                   "Sucesso", JOptionPane.INFORMATION_MESSAGE);
}
catch(CentroSiglaGrande e){
    JOptionPane.showMessageDialog(null, "A sigla pode ter no máximo 10 caracteres!",
                                   "Erro", JOptionPane.ERROR_MESSAGE);
}
catch(CentroNomeGrande e){
    JOptionPane.showMessageDialog(null, "O nome pode ter no máximo 50 caracteres!",
                                   "Erro", JOptionPane.ERROR_MESSAGE);
}
catch(CentroCamposNaoInformados e){
    JOptionPane.showMessageDialog(null, "Por favor, preencha todos os campos!",
                                   "Erro", JOptionPane.ERROR_MESSAGE);
}
catch(CentroSiglaUsada e){
    JOptionPane.showMessageDialog(null, "A sigla informada já está sendo usada!",
                                   "Erro", JOptionPane.ERROR_MESSAGE);
}

```

Fonte: Arquivo pessoal, 2024.

#### 4.4.2 Funcionamento da Interface

A interface gráfica da biblioteca foi implementada utilizando o NetBeans, com foco principal na usabilidade e facilidade de entendimento pelo usuário. A interface foi projetada de maneira intuitiva, visando proporcionar uma navegação simples e eficiente. Elementos como botões de ação, campos de pesquisa e tabelas foram organizados de forma clara, permitindo que os usuários possam adicionar, consultar e remover dados com facilidade. O design priorizou a experiência do usuário, garantindo que as funcionalidades do sistema sejam acessíveis e compreensíveis, mesmo para aqueles com pouca familiaridade com tecnologias. Em seguida será ilustrada e explicada a funcionalidade de cada janela da interface.

##### 4.4.2.1 Cadastrar Dados nas Tabelas

Para cadastrar um novo dado na tabela, basta preencher os campos à esquerda da entidade escolhida e clicar no botão Cadastrar, conforme a Figura 44. Cada



aba na interface possui os dados de cada entidade do modelo, à esquerda sempre terá os campos dos atributos e relacionamentos de cada entidade para serem preenchidos.

Figura 44 – Exemplificação de Cadastro de Dados nas Tabelas

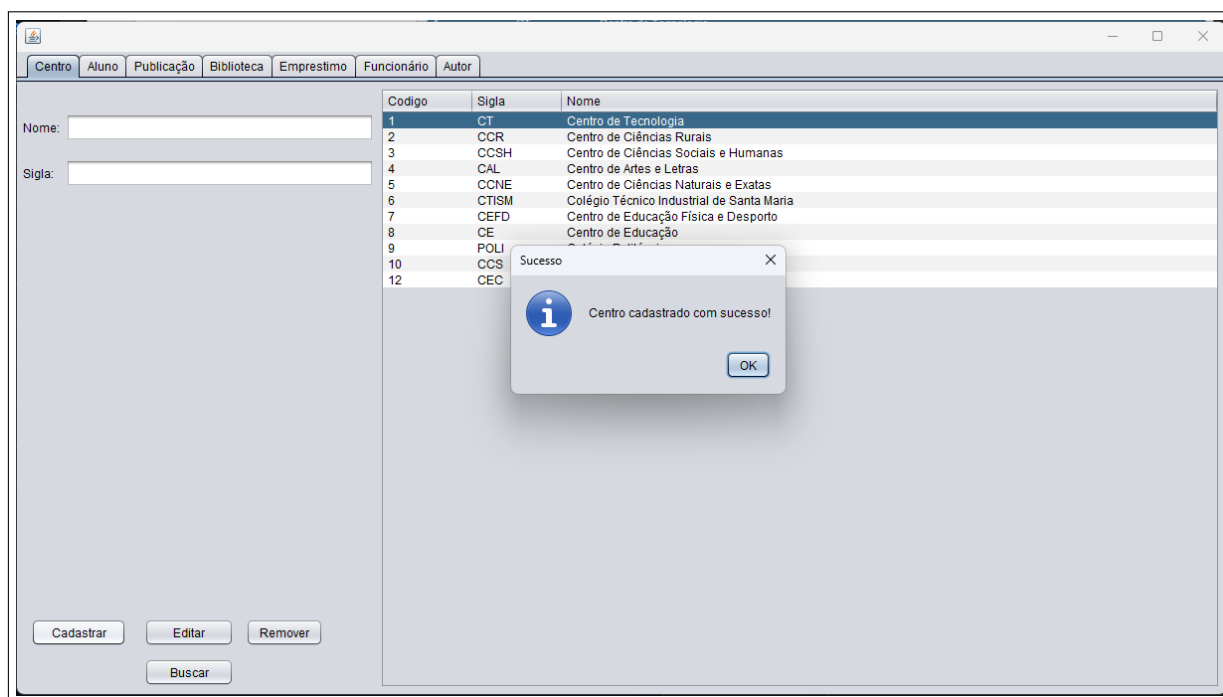
Codigo	Sigla	Nome
1	CT	Centro de Tecnologia
2	CCR	Centro de Ciências Rurais
3	CCSH	Centro de Ciências Sociais e Humanas
4	CAL	Centro de Artes e Letras
5	CCNE	Centro de Ciências Naturais e Exatas
6	CTISM	Colégio Técnico Industrial de Santa Maria
7	CEFD	Centro de Educação Física e Desporto
8	CE	Centro de Educação
9	POLI	Colégio Politécnico
10	CCS	Centro de Ciências da Saúde

Fonte: Arquivo pessoal, 2024.

Após clicar no botão Cadastrar, se todos os dados estiverem preenchidos corretamente aparecerá uma mensagem em uma caixa de texto confirmando o cadastro conforme ilustrado na Figura 45. Caso algum valor tenha sido preenchido de forma

irregular, ou não tenha sido preenchido aparecerá uma mensagem de aviso.

Figura 45 – Exemplificação de Conclusão do Cadastro de Dados nas Tabelas



Fonte: Arquivo pessoal, 2024.

Depois do aparecimento da mensagem de sucesso, após clicar em *OK*, os dados cadastrados já poderão ser observados na tabela. O código do dado cadastrado será preenchido automaticamente de acordo com a função auto-incremento. A figura

46 ilustra esse processo.

Figura 46 – Exemplificação de Visualização do Dado Cadastrado na Tabela

Codigo	Sigla	Nome
1	CT	Centro de Tecnologia
2	CCR	Centro de Ciências Rurais
3	CCSH	Centro de Ciências Sociais e Humanas
4	CAL	Centro de Artes e Letras
5	CCNE	Centro de Ciências Naturais e Exatas
6	CTISM	Colégio Técnico Industrial de Santa Maria
7	CEFD	Centro de Educação Física e Desporto
8	CE	Centro Esportivo
9	POLI	Colégio Politécnico
10	CCS	Centro de Ciências da Saúde
13	CEC	Centro de Engenharia de Computação

Fonte: Arquivo pessoal, 2024.

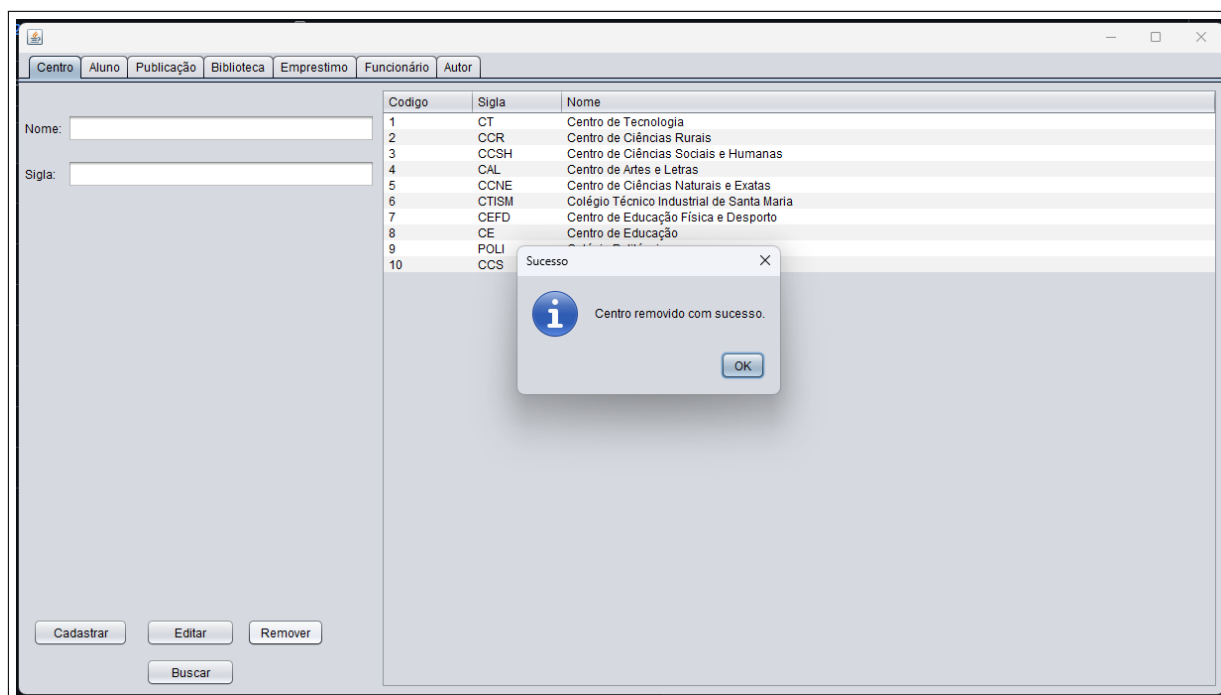
O mesmo passo a passo é aplicado para o cadastro de dados nas demais janelas.

#### 4.4.2.2 Remover Dados das Tabelas

Para remover os Dados da tabela, basta selecionar a linha da tabela que deseja remover e clicar no botão Remover, que aparecerá uma caixa de texto indicando o sucesso da remoção. A Figura 47 demonstra a remoção dos Dados a pouco cadas-

trados na subseção anterior 4.4.2.1.

Figura 47 – Exemplificação de Remoção de Dado da Tabela



Fonte: Arquivo pessoal, 2024.

#### 4.4.2.3 Editar Dados das Tabelas

Para editar algum dado da tabela, basta selecionar a linha desejada, que todas as informações da linha serão carregadas nos campos à esquerda, e então editá-los.

conforme necessidade. Conforme ilustra a Figura 48

Figura 48 – Exemplificação de Edição de Dados da Tabela

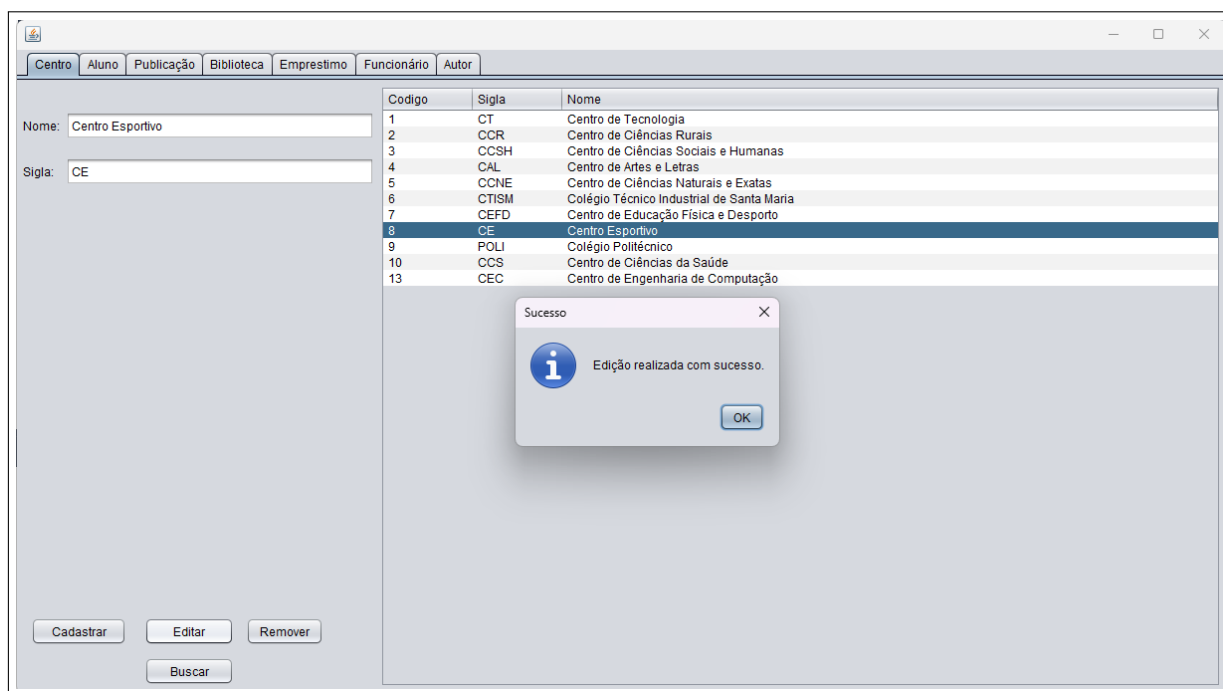
Codigo	Sigla	Nome
1	CT	Centro de Tecnologia
2	CCR	Centro de Ciências Rurais
3	CCSH	Centro de Ciências Sociais e Humanas
4	CAL	Centro de Artes e Letras
5	CCNE	Centro de Ciências Naturais e Exatas
6	CTISM	Colégio Técnico Industrial de Santa Maria
7	CEFD	Centro de Educação Física e Desporto
8	CE	Centro de Educação
9	POLI	Colégio Politécnico
10	CCS	Centro de Ciências da Saúde

Fonte: Arquivo pessoal, 2024.

Depois de concluir a edição, basta clicar no botão editar que os dados serão

atualizados na tabela e no Banco de Dados, exemplificado na Figura 49.

Figura 49 – Exemplificação de Conclusão da Edição de Dados da Tabela



Fonte: Arquivo pessoal, 2024.

#### 4.4.2.4 Buscar Dados nas Tabelas

As buscas podem ser realizadas por qualquer atributo ou relacionamento específico, em separado ou em conjunto, as Figuras 50 e 51 exemplificam essa funcio-

nalidade.

Figura 50 – Exemplificação de Busca de Dados da Tabela

Codigo	Sigla	Nome
3	CCSH	Centro de Ciências Sociais e Humanas

Fonte: Arquivo pessoal, 2024.

Figura 51 – Exemplificação de Busca de Dados da Tabela

Codigo	Sigla	Nome
1	CT	Centro de Tecnologia

Fonte: Arquivo pessoal, 2024.

#### 4.4.2.5 Janela de Centros

As Janelas já iniciam carregadas com os dados do Banco de Dados referente a cada entidade. A figura 52 ilustra a organização das entidades centros, que possuem os atributos Nome e Sigla.

Figura 52 – Janela de Dados dos Centros

Codigo	Sigla	Nome
1	CT	Centro de Tecnologia
2	CCR	Centro de Ciências Rurais
3	CCSH	Centro de Ciências Sociais e Humanas
4	CAL	Centro de Artes e Letras
5	CCNE	Centro de Ciências Naturais e Exatas
6	CTISM	Colégio Técnico Industrial de Santa Maria
7	CEFD	Centro de Educação Física e Desporto
8	CE	Centro Esportivo
9	POLI	Colégio Politécnico
10	CCS	Centro de Ciências da Saúde

Fonte: Arquivo pessoal, 2024.

Caso existam Alunos cadastrados em determinado Centro, a remoção se torna impossível, e uma caixa de texto indica a existência de alunos que devem ser removidos do centro antes da remoção.

#### 4.4.2.6 Janela de Alunos

A Janela de Alunos possui os campos de Matrícula, Nome, Endereço que são atributos e Centro, que representa o relacionamento. A aba Centro possui todos os centros cadastrados até o momento. Para cadastrar um aluno é necessário o associar



a um desses Centros. A Figura 53 ilustra essa janela.

Figura 53 – Janela de Dados de Alunos

Matricula	Nome	Centro	Endereço
202220000	José Mourinho	CT	Avenida Augusta
202220001	Carlos Ancelotti	CCR	Avenida Paulista
202220002	Pep Guardiola	CCSH	Rua das Flores
202220003	Roger Machado	CAL	Rua do Sol
202220004	Zinedine Zidane	CCNE	Avenida Brasil
202220005	Diego Simeone	CCSH	Rua da Liberdade
202220006	Renato Gaúcho	CEFD	Rua do Porto
202220007	Antonio Conte	CE	Rua Rio de Janeiro
202220008	Arsène Wenger	CCSH	Avenida São João
202220009	Frank Lampard	CCS	Rua do Carmo
202220010	David Moyes	CTISM	Rua das Palmeiras
202220011	Claudio Ranieri	CCR	Avenida dos Eucaliptos
202220012	Marcelo Bielsa	CCSH	Rua da Paz
202220013	Tite	CAL	Rua do Parque
202220014	Luis Enrique	CCNE	Avenida do Mar

Fonte: Arquivo pessoal, 2024.

#### 4.4.2.7 Janela de Publicações

A Janela de Publicação possui os atributos Nome, Ano e o Relacionamento Biblioteca, que funciona da mesma forma que o relacionamento Centro da Janela Alunos da Secção 4.4.2.6. A entidade Publicação possui uma especialização, portanto ao adicionar uma nova Publicação a Tabela, é necessário indicar o Tipo de Publicação e

preencher o atributo correspondente. A figura 54 ilustra essa Janela.

Figura 54 – Janela de Dados de Publicação

Codigo	Nome	Ano	Biblioteca	Tipo
1	Introdução à Ciência de Dados	2020	BC	Acadêmico
2	Matemática Avançada	2019	BSCT	Acadêmico
3	Fundamentos de Engenharia	2018	BSCCR	Acadêmico
4	Romance Clássico	2021	BSCCSH	Literatura
5	O Poder do Hábito	2022	BSCAL	Autoajuda
6	O Senhor dos Anéis	2017	BSCCNE	Literatura
7	Física para Engenheiros	2020	BSCTISM	Acadêmico

Fonte: Arquivo pessoal, 2024.

#### 4.4.2.8 Janela de Bibliotecas

A Janela Biblioteca possui os atributos Nome, Endereço e Sigla e um relacionamento com um centro cadastrado. A Figura 55 ilustra essa janela.

Figura 55 – Janela de Dados de Bibliotecas

Codigo	Sigla	Nome	Endereço
1	BC	Biblioteca Central	Street 1
2	BSCT	Biblioteca Setorial do Centro de Tecnologia	Street 2
3	BSCCR	Biblioteca Setorial do Centro de Ciências Rurais	Street 3
4	BSCCSH	Biblioteca Setorial do Centro de Ciências Sociais e Humanas	Street 4
5	BSCAL	Biblioteca Setorial do Centro de Artes e Letras	Street 5
6	BSCCNE	Biblioteca Setorial do Centro de Ciências Naturais e Exatas	Street 6
7	BSCTISM	Biblioteca Setorial do Centro do Colégio Técnico Industrial de Santa Maria	Street 7
8	BSCEFD	Biblioteca Setorial do Centro de Educação Física e Desporto	Street 8
9	BSCE	Biblioteca Setorial do Centro de Educação	Street 9
10	BSPOLI	Biblioteca Setorial do Centro do Colégio Politécnico	Street 10
11	BSCCS	Biblioteca Setorial do Centro de Ciências da Saúde	Street 11

Fonte: Arquivo pessoal, 2024.

#### 4.4.2.9 Janela de Empréstimos

A janela Empréstimo possui os atributos Data e Hora para preenchimento e relacionamentos com o Funcionário que realizou o empréstimo, publicação emprestada

e Aluno que solicitou o empréstimo. A Figura 56 ilustra essa janela.

Figura 56 – Janela de Dados de Empréstimos

Codigo	Data	Hora	Funcionário	Publicação	Aluno
1	2024-01-12	14:30:00	Lionel Messi	Introdução à Ciência de Dados	José Mourinho
2	2024-01-13	10:00:00	Cristiano Ronaldo	Matemática Avançada	Carlos Ancelotti
3	2024-01-14	16:15:00	Neymar Jr.	Fundamentos de Engenharia	Pep Guardiola
4	2024-02-20	11:20:00	Kylian Mbappé	Romance Clássico	Roger Machado
5	2024-02-21	14:45:00	Luka Modrić	O Poder do Hábito	Zinedine Zidane
6	2024-03-05	09:30:00	Sergio Ramos	O Senhor dos Anéis	Diego Simeone
7	2024-03-10	13:00:00	Zlatan Ibrahimović	Física para Engenheiros	Renato Gaúcho

Fonte: Arquivo pessoal, 2024.

#### 4.4.2.10 Janela de Funcionários

A janela Funcionário possui os atributos nome e salário, além disso, possui os campos de Salário Mínimo, Salário Máximo e Média Salarial, calculados utilizando as

funções MAX, MIN e AVG da linguagem MariaDB. A Figura 57 ilustra essa janela.

Figura 57 – Janela de Dados de Funcionários

Codigo	Nome	Salário
1	Lionel Messi	9398.98
2	Cristiano Ronaldo	10000.00
3	Neymar Jr.	9500.50
4	Kylian Mbappé	8700.75
5	Luka Modrić	11000.00
6	Sergio Ramiro	9800.20
7	Zlatan Ibrahimović	9600.35
8	Kevin De Bruyne	9200.60
9	Mohamed Salah	9400.80
10	Robert Lewandowski	10500.90
11	Virgil van Dijk	9800.40
12	Harry Kane	9150.30
14	Erling Haaland	9250.00
15	Enner Valência	10050.60
18	Harry Kane	1230.00

Fonte: Arquivo pessoal, 2024.

#### 4.4.2.11 Janela de Autores

A janela Autor possui os atributos Nome e País, a Figura 58 ilustra essa janela.

Figura 58 – Janela de Dados de Autores

Codigo	Nome	País
1	J.K. Rowling	Reino Unido
2	George Orwell	Reino Unido
3	Clarice Lispector	Brasil
4	Gabriel García Márquez	Colômbia
5	Leo Tolstoy	Rússia
6	Carlos Drummond	Brasil
7	Jane Austen	Reino Unido
8	Mark Twain	Estados Unidos
9	Machado de Assis	Brasil
10	Ernest Hemingway	Estados Unidos

Fonte: Arquivo pessoal, 2024.

## 5 Conclusão

O desenvolvimento deste projeto, que envolveu desde a modelagem conceitual até a implementação física e criação de uma interface gráfica, proporcionou uma compreensão ampla e prática do processo de construção de um banco de dados. Inicialmente, a criação dos Modelos Conceitual e Lógico destacou a importância de estruturar e organizar as informações de forma eficiente, a fim de garantir a integridade e escalabilidade do sistema. Esta etapa inicial consolidou os fundamentos teóricos aprendidos ao longo da disciplina e preparou a base para as etapas subsequentes.

Na etapa seguinte, a transição para o modelo físico na linguagem SQL MariaDB permitiu a aplicação dos conceitos em um ambiente de banco de dados real. A criação de tabelas e a manipulação de comandos SQL demonstraram a relevância da modelagem adequada, bem como a importância de otimizar o desempenho e assegurar a consistência dos dados. A integração deste banco de dados com uma interface gráfica funcional, que permitiu adicionar, remover e consultar elementos, foi fundamental para tornar o sistema acessível e prático para o usuário final.

O projeto também contribuiu significativamente para o desenvolvimento de habilidades técnicas, como o uso de ferramentas de modelagem, a escrita de relatórios seguindo normas acadêmicas e a programação em SQL. Além disso, a criação da interface gráfica proporcionou uma experiência prática na integração de diferentes tecnologias, abordando desafios como o tratamento de exceções e a usabilidade do sistema.

Por fim, o trabalho incentivou o desenvolvimento de competências interpessoais, como a colaboração em equipe e a comunicação eficaz entre os membros do grupo. A experiência proporcionada por este projeto nos preparou para enfrentar futuros desafios profissionais, consolidando o aprendizado teórico e prático na área de banco de dados e sistemas de informação. Dessa forma, conclui-se que os objetivos gerais foram plenamente atingidos, resultando em um sistema funcional e integrado que reflete a aplicação prática dos conceitos estudados.

## Referências Bibliográficas

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009.

SCHULTZ, Vinícius et al. **grupo7-CDI-20242-tg2**. 2024. Disponível em: <<https://github.com/VenixBR/grupo7-CDI-20242-tg2/tree/main>>.