

JOSÉ CARLOS DOS SANTOS (20200150373)

**DOCUMENTAÇÃO DE REQUISITOS PARA SISTEMA GERENCIAMENTO DE
EMPRÉSTIMOS**

Sumário

Prefácio	2
1 Introdução	3
2 Modelo Conceitual	3
2.1 Painel é, não é	3
2.2 Estórias do Usuário	4
3 Modelo lógico	5
3.1 Normalização	5
3.1.1 Relação empréstimo	5
3.1.1.1 Conversão para a primeira forma normal	5
3.1.1.2 Conversão para a segunda forma normal	6
3.1.1.3 Conversão para a terceira forma normal	6
3.1.2 Relação livro	7
3.1.2.1 Conversão para a primeira forma normal	8
3.1.2.2 Conversão para a segunda forma normal	8
3.1.2.3 Conversão para a terceira forma normal	9
3.2 Diagrama de classes UML	10
4. Modelo Físico	11
4.1 Esquema do banco de dados contendo as tabelas	11
4.2 Criação das estruturas de manipulação e consulta de informações	12
5 Apêndices	15
5.1 Regras de negócio	15

Prefácio

Este documento tem a finalidade de apresentar um detalhamento dos requisitos do projeto de gerenciamento de uma biblioteca, com o uso da ferramenta MySQL Workbench e por meio da linguagem SQL. O projeto possibilitará o usuário e administradores do sistema fazerem uso de suas funcionalidades.

1 Introdução

As bibliotecas de livros costumam realizar muitos empréstimos para seus usuários. Ações como cadastrar usuário, cadastrar livros e informações sobre livros, empréstimos e devoluções estão presentes nas rotinas administrativas da biblioteca e gerenciar todos esses atributos e ações exige muita organização. Nesse sentido, as aplicações e os sistemas de bancos de dados podem facilitar todo o processo de gerenciamento de livros, usuários, empréstimos etc.

O presente projeto tem a finalidade de cumprir esse gerenciamento por meio de um banco de dados, projetado para armazenar e atender as demandas citadas acima. Por meio de funções como cadastrar usuários, cadastrar livros e áreas do conhecimento, cadastrar editoras e autores, consultar livros e fazer empréstimos, o sistema projetado irá cumprir rotinas básicas de uma biblioteca.

2 Modelo Conceitual

2.1 Painel é, não é

É <ul style="list-style-type: none">• Gerenciador de empréstimos de livros a usuários• Acervo de livros disponível para consulta aos usuários com ou sem cadastro	NÃO É <ul style="list-style-type: none">• Acervo de dados pessoais de usuários• Acervo de dados pessoais de autores
FAZ <ul style="list-style-type: none">• Cadastro de livros e autores• Cadastro de área do conhecimento• Cadastro de usuários• Empréstimos de livros• Consulta de livros e autores• Cadastro de editoras	NÃO FAZ <ul style="list-style-type: none">• Renova empréstimos• Empréstimos de vários livros por usuário

Figura 1 - Painel é não é

2.2 Estórias do Usuário

Atores: Usuário e administrador do sistema

Tabela 1 – Estórias do Usuário

EU1	Como usuário preciso realizar um cadastro de informações pessoais.
EU2	Como administrador do sistema devo ter o controle para cadastrar obras.
EU3	Como usuário preciso consultar as obras da biblioteca
EU4	Como usuário preciso fazer empréstimos de obras da biblioteca
EU5	Como administrador do sistema devo ter o controle para cadastrar informações das obras

3 Modelo lógico

3.1 Normalização


Após a criação de alguns exemplos para análise, três relações foram encontradas e duas delas necessitaram ser normalizadas. São elas, a tabela **usuário**, **empréstimo** e **livro**.


3.1.1 Relação empréstimo

Tabela 2 – Relação empréstimo

CPF	idLivro	DataEmprestimo	DataPrevistaDevolucao	DataDevolucao
35	10	25/05/18	09/06/18	07/06/18
79	1	28/03/19	12/04/19	13/04/19
35	12	03/05/20	18/05/20	17/05/20
53	4	10/08/21	25/08/21	25/08/21
35	10	25/11/18	09/12/18	07/12/18
67	12	03/05/20	18/05/20	17/05/20

3.1.1.1 Conversão para a primeira forma normal

Passo 1: Eliminar os grupos repetitivos 

Passo 2: Identificar a chave principal 

Uma vez que não é possível um mesmo usuário fazer dois empréstimos e que este é limitado a um empréstimo por dia, os atributos dos qual identifica uma única tupla na relação Empréstimo é o CPF e DataEmprestimo. Portanto:

(CPF, DataEmprestimo, idLivro, DataPrevistaDevolucao, DataDevolucao)

Passo 3: Identificar todas as dependências

a) Dependências parciais

(**DataEmprestimo**, DataPrevistaDevolucao, DataDevolucao)

(**CPF**, **DataEmprestimo**, idLivro)

b) Dependências transitivas

Não há dependências transitivas.

3.1.1.2 Conversão para a segunda forma normal

Periodo (**DataEmprestimo**, DataPrevistaDevolucao, DataDevolucao)

Transferência (**CPF**, **DataEmprestimo**, idLivro)

3.1.1.3 Conversão para a terceira forma normal

Passo 1: Fazer novas tabelas para eliminar dependências transitivas

Não houve dependências transitivas. Logo, as relações criadas na 2FN já estão na 3FN. As novas relações são:

Periodo (**DataEmprestimo**, DataPrevistaDevolucao, DataDevolucao)


Transferência (**CPF**, **DataEmprestimo**, idLivro)


3.1.2 Relação livro

Tabela 3 – Relação livro

IdLivro	ISBN	Título	IdAutor	Autor	IdAreaConhecimento	AreaConhecimento	Edicao	idEditora	Editora	AnoPublicacao	PaisPublicacao	StatusLivro
1	25141	O Garoto Que Nao Conseguia Dormir	5	Elis Regina	1	Drama	12	1	Esperanca	1998	Espanha	Disponivel
4	74585	Minha doce Aprendiz	2	Catarina Dias	1	Drama	7	1	Esperanca	2001	Estados Unidos	Disponivel
10	74585	Minha doce Aprendiz	2	Catarina Dias	1	Drama	7	1	Esperanca	2001	Estados Unidos	Disponivel
11	25141	O Garoto Que Nao Conseguia Dormir	5	Elis Regina	1	Drama	12	1	Esperanca	1998	Espanha	Disponivel
12	74585	Minha doce Aprendiz	2	Catarina Dias	1	Drama	7	1	Esperanca	2001	Estados Unidos	Disponivel
15	25141	O Garoto Que Nao Conseguia Dormir	5	Elis Regina	1	Drama	12	1	Esperanca	1998	Espanha	Disponivel
2	63524	Os Deuses Eram Astronautas	3	Julio Vernes	2	Ficcao Cientifica	5	4	Terra	2000	Inglaterra	Indisponivel
3	98545	Alienigenas do Passado	3	Julio Vernes	2	Ficcao Cientifica	2	4	Terra	1986	Italia	Disponivel
5	96655	Ovnis: A Chegada	1	Einstein	2	Ficcao Cientifica	10	2	Astros	1992	Alemanha	Indisponivel
7	96655	Ovnis: A Chegada	1	Einstein	2	Ficcao Cientifica	10	2	Astros	1992	Alemanha	Indisponivel
8	98545	Alienigenas do Passado	3	Julio Vernes	2	Ficcao Cientifica	2	4	Terra	1986	Italia	Disponivel
9	63524	Os Deuses Eram Astronautas	3	Julio Vernes	2	Ficcao Cientifica	5	4	Terra	2000	Inglaterra	Indisponivel
13	63524	Os Deuses Eram Astronautas	3	Julio Vernes	2	Ficcao Cientifica	5	4	Terra	2000	Inglaterra	Indisponivel
14	98545	Alienigenas do Passado	3	Julio Vernes	2	Ficcao Cientifica	2	4	Terra	1986	Italia	Disponivel
17	96655	Ovnis: A Chegada	1	Einstein	2	Ficcao Cientifica	10	2	Astros	1992	Alemanha	Indisponivel
6	21556	Versos Para Minha Amada	4	Machado de Assis	3	Romance	19	3	Vida	1998	Brasil	Indisponivel
16	21556	Versos Para Minha Amada	4	Machado de Assis	3	Romance	19	3	Vida	1998	Brasil	Indisponivel

3.1.2.1 Conversão para a primeira forma normal

Passo 1: Eliminar os grupos repetitivos 

Passo 2: Identificar a chave principal 

O atributo do qual identifica uma única tupla na relação Livro é o idLivro. Portanto:

Livro (**idLivro**, ISBN, Titulo, IdAutor, Autor, IdAreaConhecimento, AreaConhecimento, Edicao, idEditora, Editora, AnoPublicacao, PaisPublicacao, StatusLivro)

Passo 3: Identificar todas as dependências

a) Dependências parciais

Não há

b) Dependências transitivas

IdAutor → Autor

IdAreaConhecimento → AreaConhecimento

idEditora → Editora

Como não há dependências parciais, em virtude de não haver chave primária composta, a relação Livro já está na 1FN.

3.1.2.2 Conversão para a segunda forma normal

Como não houve dependências, as relações que eliminam as dependências não serão criadas. Ou seja, a relação Livro já se encontra na 2FN.

3.1.2.3 Conversão para a terceira forma normal

Passo 1: Fazer novas tabelas para eliminar dependências transitivas

(**IdAutor**, Autor)

(**IdAreaConhecimento**, AreaConhecimento)

(**IdEditora**, Editora)

Os atributos Autor, AreaConhecimento e Editora foram renomeados para NomeAutor, Descricao e NomeEditora, respectivamente. Assim sendo, as novas relações ficam da seguinte forma:

Autor (**IdAutor**, NomeAutor)

Área (**IdAreaConhecimento**, Descricao)

Editora (**IdEditora**, NomeEditora)

As novas relações são:

Livro (**idLivro**, ISBN, Titulo, Edicao, AnoPublicacao, PaisPublicacao, StatusLivro, IdAutor, IdAreaConhecimento, IdEditora)

Autor (**IdAutor**, NomeAutor)

Área (**IdAreaConhecimento**, Descricao)

Editora (**IdEditora**, NomeEditora)

O atributo Edicao foi alterado para NumEdicao. As novas relações são:

Livro (**idLivro**, ISBN, Titulo, StatusLivro, IdAreaConhecimento, NumEdicao, AnoPublicacao, PaisPublicacao, IdAutor)

Autor (**IdAutor**, NomeAutor)

Área (**IdAreaConhecimento**, Descricao)

Editora (**IdEditora**, NomeEditora)

3.2 Diagrama de classes UML

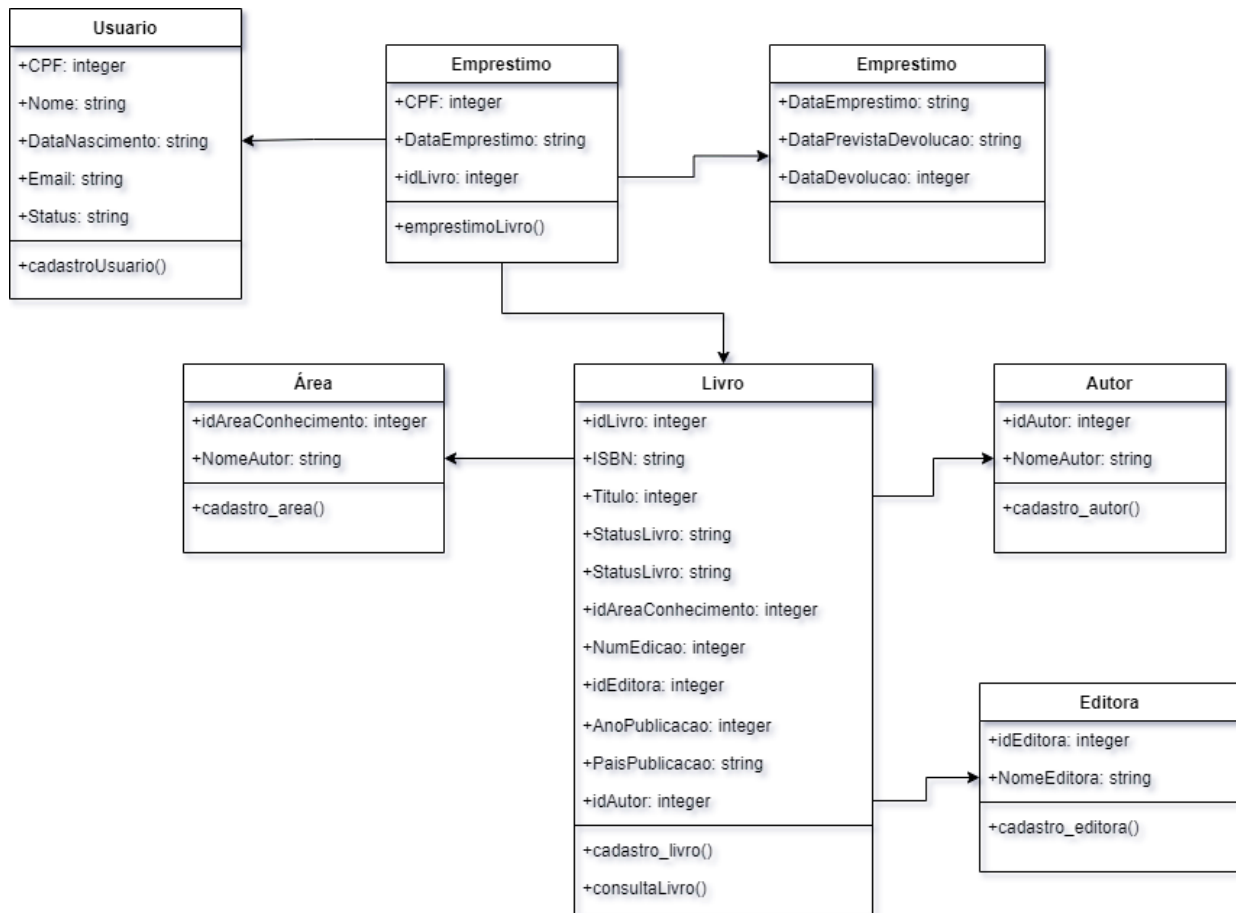


Figura 2 - Diagrama de classes

4. Modelo Físico

4.1 Esquema do banco de dados contendo as tabelas

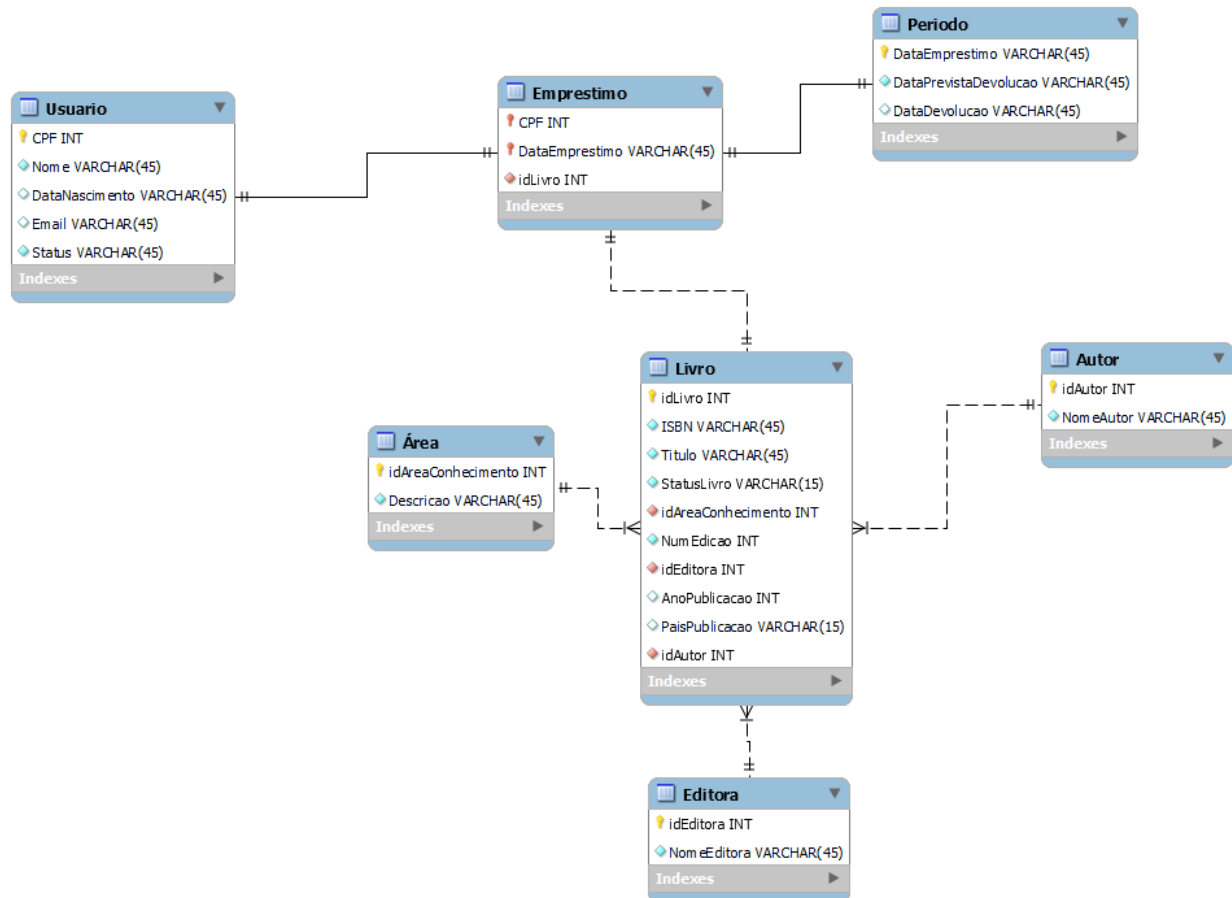


Figura 3 - Esquema de tabelas

4.2 Criação das estruturas de manipulação e consulta de informações

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cadastro_area`(IN Nome_Area VARCHAR(45))
BEGIN
    IF NOT EXISTS (
        SELECT Descricao
        FROM mydb.área
        WHERE Descricao = Nome_Area
    ) THEN
        INSERT INTO `mydb`.`área` (`Descricao`) VALUES (Nome_Area);
    END IF;
END
```

Figura 4 - Procedure cadastro_area

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cadastro_autor`(IN Nome_Autor VARCHAR(45))
BEGIN
    IF NOT EXISTS (
        SELECT NomeAutor
        FROM mydb.autor
        WHERE NomeAutor = Nome_Autor
    ) THEN
        INSERT INTO `mydb`.`autor` (`NomeAutor`) VALUES (Nome_Autor);
    END IF;
END
```

Figura 5 - Procedure cadastro_autor

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cadastro_editora`(IN Nome_Editora VARCHAR(45))
BEGIN
    IF NOT EXISTS (
        SELECT NomeEditora
        FROM mydb.editora
        WHERE NomeEditora = Nome_Editora
    ) THEN
        INSERT INTO `mydb`.`editora` (`NomeEditora`) VALUES (Nome_Editora);
    END IF;
END
```

Figura 6 - Procedure cadastro_editora

Houve outras procedures criadas para as rotinas que não foram expostas em virtude de seu tamanho. As procedures das figuras acima e as que faltaram encontram-se no arquivo de banco de dados, anexo neste documento.

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `mydb`.`consultaLivro` AS
  SELECT
    `mydb`.`livro`.`idLivro` AS `idLivro`,
    `mydb`.`livro`.`ISBN` AS `ISBN`,
    `mydb`.`livro`.`Titulo` AS `Titulo`,
    `mydb`.`livro`.`StatusLivro` AS `StatusLivro`,
    `mydb`.`livro`.`idAreaConhecimento` AS `idAreaConhecimento`,
    `mydb`.`livro`.`NumEdicao` AS `NumEdicao`,
    `mydb`.`livro`.`idEditora` AS `idEditora`,
    `mydb`.`livro`.`AnoPublicacao` AS `AnoPublicacao`,
    `mydb`.`livro`.`PaisPublicacao` AS `PaisPublicacao`,
    `mydb`.`livro`.`idAutor` AS `idAutor`
  FROM
    `mydb`.`livro`
  WHERE
    (`mydb`.`livro`.`idLivro` = 20)
```

Figura 7 - View consultaLivro

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `mydb`.`consultaarea` AS
  SELECT
    `mydb`.`area`.`idAreaConhecimento` AS idAreaConhecimento,
    `mydb`.`area`.`Descricao` AS Descricao
  FROM
    `mydb`.`area`
```

Figura 8 - view consultaarea

4.3 Inserção de dados de exemplo no banco

Os comandos abaixo representam inserções de dados nas relações presente no banco de dados.

```
-----
-- Data for table `mydb`.`Área`
-----

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`Área` (`idAreaConhecimento`, `Descricao`) VALUES (1, 'Drama');
INSERT INTO `mydb`.`Área` (`idAreaConhecimento`, `Descricao`) VALUES (2, 'Ficcao Cientifica');
INSERT INTO `mydb`.`Área` (`idAreaConhecimento`, `Descricao`) VALUES (3, 'Romance');
COMMIT;

-----
-- Data for table `mydb`.`Editora`
-----

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`Editora` (`idEditora`, `NomeEditora`) VALUES (1, 'Esperanca');
INSERT INTO `mydb`.`Editora` (`idEditora`, `NomeEditora`) VALUES (2, 'Astros');
INSERT INTO `mydb`.`Editora` (`idEditora`, `NomeEditora`) VALUES (3, 'Vida');
INSERT INTO `mydb`.`Editora` (`idEditora`, `NomeEditora`) VALUES (4, 'Terra');
COMMIT;

-----
-- Data for table `mydb`.`Autor`
-----

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`Autor` (`idAutor`, `NomeAutor`) VALUES (1, 'Einstein');
INSERT INTO `mydb`.`Autor` (`idAutor`, `NomeAutor`) VALUES (2, 'Catarina Dias');
INSERT INTO `mydb`.`Autor` (`idAutor`, `NomeAutor`) VALUES (3, 'Julio Vernes');
INSERT INTO `mydb`.`Autor` (`idAutor`, `NomeAutor`) VALUES (4, 'Machado de Assis');
INSERT INTO `mydb`.`Autor` (`idAutor`, `NomeAutor`) VALUES (5, 'Elis Regina');
COMMIT;

-----
-- Data for table `mydb`.`Livro`
-----

START TRANSACTION;
USE `mydb`;
```

```
INSERT INTO `mydb`.`Livro` (`idLivro`, `ISBN`, `Titulo`, `StatusLivro`, `idAreaConhecimento`, `NumEdicao`, `idEditora`,  
`AnoPublicacao`, `PaisPublicacao`, `idAutor`) VALUES (1, '25141', 'O Garoto Que Nao Conseguia Dormir', 'Disponivel', 1,  
12, 1, 1998, 'Espanha', 5);
```

```
-----  
-- Data for table `mydb`.`Usuario`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Usuario` (`CPF`, `Nome`, `DataNascimento`, `Email`, `Status`) VALUES (35, 'Marcos',  
'25/12/2015', 'marcos.araujo2022@hotmail.com', 'Suspenso');
```

5 Apêndices

5.1 Regras de negócio

As funcionalidades presentes no projeto incorporam regras de negócio que permitem o correto funcionamento do banco de dados. O cadastro de **livros é** possível quando o usuário não está cadastrado. O cadastro de livros, autores, áreas de conhecimento e editoras seguem o mesmo padrão.