



**UNINABUCO**

ANÁLISE E DESENVOLVIMENTO DE SOFTWARE

TÓPICOS INTEGRADORES II

# Documentação de Projeto de Software

## Versão 4.0

Autor(a): Jefferson Queiroz Pereira de Melo

Link do Github: <https://github.com/1989jeffesonqueiroz/Sistema-de-Biblioteca>

2020

## Controle de Versão do Documento

Data	Versão	Descrição
30/03/2020	1.0	Elaboração da parte teórica do projeto, bem como dos modelos: de visão, lógico, físico, conceitual e do dicionário de dados.
07/04/2020	2.0	Alteração no Modelo ER, Dicionário de dados, Modelo físico. Foram retiradas as tabelas: <b>Livro_editora</b> e <b>Livro_Autor</b> , acrescentado chaves estrangeira na tabela livro.
27/04/2020	3.0	Exclusão da tabela ADM do modelo lógico e dicionário de dados. Exclusão dos atributos da entidade ADM do modelo conceitual. Retificação do Modelo de visão. Mudança das chaves estrangeiras de id_adm para id_login nas entidades: Livro, Editora, autor, leitor e funcionário. Foi acrescentado um atributo <b>Tipo</b> na entidade Login.
19/05/2020	4.0	Inclusão do Diagrama de Classes.

# Sumário

1. Introdução ao Documento .....	4
1.1. Área de negócio do Sistema .....	4
1.2. Principais funcionalidades.....	4
1.3. Método de trabalho .....	4
2. Modelo de Dados .....	5
2.1. Modelo de Visão .....	5
2.2. Modelo Conceitual.....	6
2.3. Modelo Lógico .....	6
2.4. Dicionário de Dados .....	8
2.5. Modelo Físico .....	11
2.5.1. <i>Scripts</i> DDL do banco de dados. ....	11
2.5.2. Comandos <i>Insert</i> .....	13
2.5.3. Relatórios .....	15
3. Análise e Design .....	16
3.1. Diagrama de Classes.....	16
4. Arquitetura do Software.....	17
4.1. Padrão de projeto .....	17
4.2. Protótipo .....	17

# 1. Introdução ao Documento

## 1.1. Área de negócio do Sistema

O presente projeto é voltado para a área educacional, em específico para as bibliotecas. O intuito é proporcionar o gerenciamento do empréstimo de livros de forma similar ao software PERGAMUM já existente nas universidades públicas do estado.

## 1.2. Principais funcionalidades

O sistema de controle de biblioteca tem como funcionalidades: cadastrar, consultar, editar e excluir: leitores, livros, autores, editoras, funcionários, extrair relatórios, determinar prazo de devolução e aplicar multa em caso de descumprimento da data limite.

## 1.3. Método de trabalho

Inicialmente foi realizado o levantamento de requisitos do sistema. Em seguida, deu-se prosseguimento à criação de um diagrama entidade relacionamento (modelo ER) visando um melhor entendimento acerca do funcionamento do banco de dados. Para tal utilizou-se a ferramenta de modelagem em banco de dados BrModelo. O projeto, Sistema de Biblioteca, foi desenvolvido com a linguagem de programação Java, cuja interação deu-se com SGBD MySQL. Este foi manipulado através do *software HeidiSQL*.

## 2. Modelo de Dados

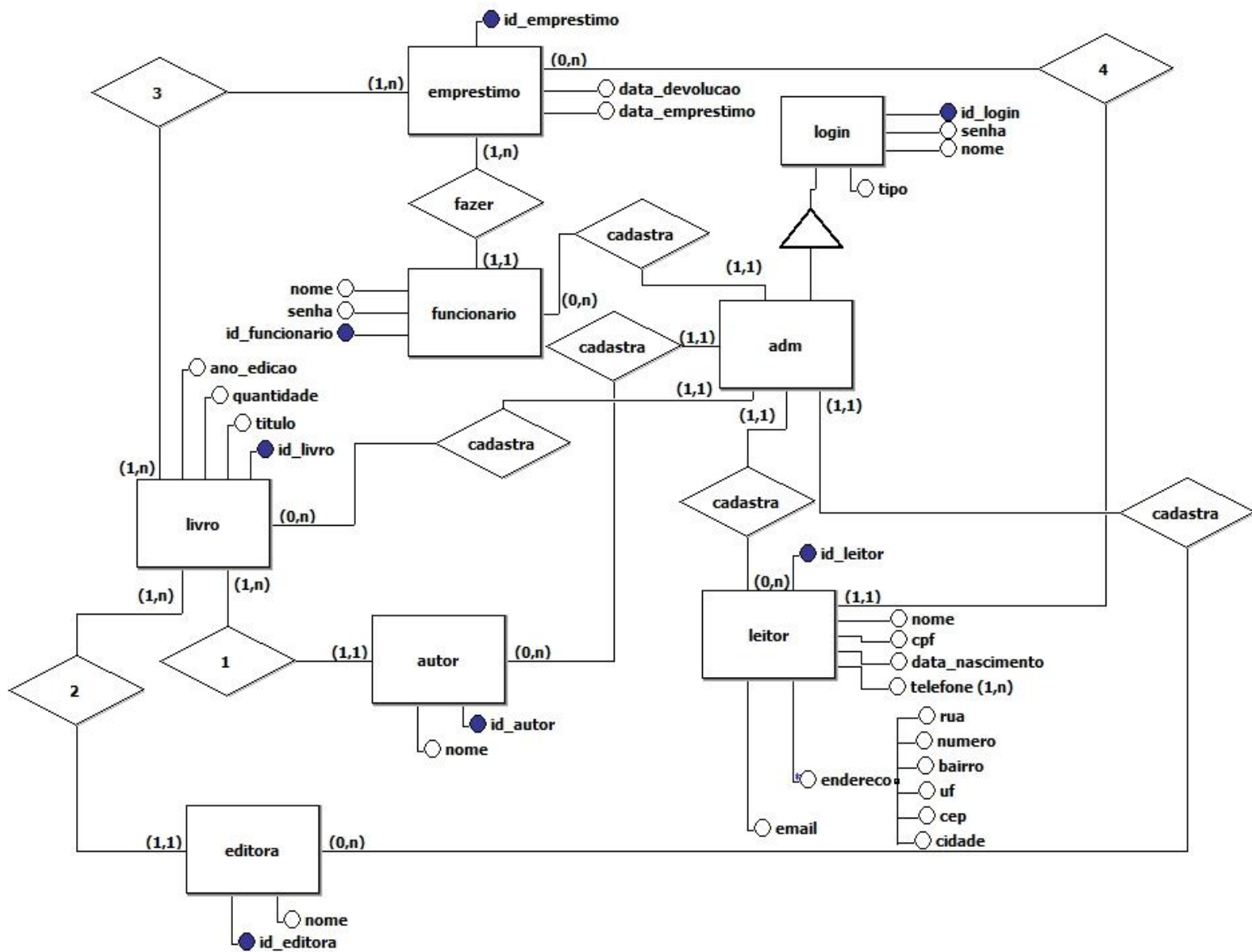
### 2.1. Modelo de Visão

- Apenas o login do tipo administrador poderá cadastrar funcionários, leitores, livros, autores, editoras.
- Os leitores só poderão ser pessoas físicas e, no cadastro, será solicitado: Nome, data de nascimento, telefone, CPF, E-mail e endereço. Ao final, se atribuirá um ID que será o *Primary Key*.
- Para cadastrar um livro, será necessário primeiramente cadastrar o autor e editora do mesmo para poder realizar o cadastro do livro, que será solicitado: Título, autor, editora, ano da edição e quantidade. Ao final, se atribuirá um ID que será o *Primary Key*.
- O empréstimo será realizado apenas pelo funcionário.
- No empréstimo, será requisitado o nome do leitor previamente cadastrado. Este escolherá um ou mais livros, será registrado a data do empréstimo e a data de devolução e será atribuído um ID que será o *Primary Key*.
- O prazo para devolução é de uma semana. Ao leitor, não será permitido a entrega de apenas um dos livros, caso tenha pego mais de um. Devolve-se tudo de uma só vez, ou renova. A opção de renovação pode ser feita por duas vezes. A cada dia de atraso, será cobrado um valor de R\$1,00.

O sistema possibilitará a extração de relatórios de várias formas:

- Relação dos leitores;
- Relação dos livros;
- Quantidade de livros emprestados por leitor;
- Quantidade de leitores que fizeram empréstimos de um determinado livro.

## 2.2. Modelo Conceitual



### 2.3. Modelo Lógico

**login** (id\_login, nome, senha, tipo)

id\_login – Primary Key

**autor** (id\_autor, nome, id\_login)

id-autor – Primary Key

id\_login referencia login - Foreign Key

**editora (id\_editora, nome, id\_login)**

id\_editora - Primary Key

id\_login referencia login - Foreign Key

**livro (id\_livro, titulo, volume, ano, id\_login, id\_editora, id\_autor)**

id\_livro - Primary Key

id\_login referencia login - Foreign Key

id\_editora referencia editora - Foreign Key

Id\_autor referencia autor - Foreign Key

**leitor (id\_leitor, nome, cpf, data\_nascimento, telefone, email, rua, número, bairro, cidade, cep, uf, id\_login)**

id\_leitor - Primary Key

id\_login referencia Login

**emprestimo (id\_emprestimo, data\_emprestimo, data\_devolucao, id\_funcionario, id\_leitor)**

id\_emprestimo - Primary Key

id\_funcionario referencia funcionário - Foreign Key

id\_leitor referencia leitor - Foreign Key

**funcionario (id\_funcionario, nome, senha, id\_login)**

id\_funcionario - Primary Key

id\_login referencia login - Foreign Key

**emprestimo livro (id\_emprestimo, id\_livro)**

id\_emprestimo referencia empréstimo - Primary Key

id\_livro referencia livro - Primary Key

## 2.4. Dicionário de Dados

**Tabela 1.** *Login*

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Nome</b>	Varchar,100	Sim		Nome do usuário que está logando
<b>Senha</b>	Varchar,50	Sim		Senha do usuário que está logando
<b>Tipo</b>	Varchar,11	Sim		Contém os dados se a conta logada é Admin ou Funcionário

**Tabela 2.** *Autor*

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Nome</b>	Varchar,100	Sim		Nome do autor
<b>Id_login</b>	Integer	Sim	FK	Referência à Tabela login

**Tabela 3.** *Livro*

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Título</b>	Varchar,100	Sim		Nome do Livro,
<b>Volume</b>	Integer	Sim		Quantidade em estoque
<b>Ano</b>	Char,4	Sim		Ano da edição do livro
<b>Id_login</b>	Integer	Sim	FK	Referencia à Tabela login
<b>Id_editora</b>	Integer	Sim	FK	Referencia à Tabela Editora
<b>Id_autor</b>	Integer	Sim	FK	Referencia à Tabela Autor



**Tabela 4.** Funcionario

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Nome</b>	Varchar,100	Sim		Nome do funcionário,
<b>Senha</b>	Varchar,50	Sim		Senha do funcionário
<b>Id_login</b>	Integer	Sim	FK	Referencia à tabela login

**Tabela 5.** Emprestimo

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Data_emprestimo</b>	Date	Sim		Data da locação do(s) Livro(s)
<b>Data_devolucao</b>	Date	Sim		Data da entrega do(s) Livro(s)
<b>Id_funcionario</b>	Integer	Sim	FK	Referencia à Tabela Funcionario
<b>Id_leitor</b>	Integer	Sim	FK	Referencia à Tabela Leitor

**Tabela 6.** Leitor

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Nome</b>	Varchar,100	Sim		Nome do Leitor
<b>Cpf</b>	Char,11	Sim		Cpf do leitor
<b>Data_nascimento</b>	Date	Sim		Data de nascimento do Leitor
<b>Telefone</b>	Char,13	Sim		Telefone do Leitor
<b>Email</b>	Varchar,100	Sim		E-mail do Leitor
<b>Rua</b>	Varchar,100	Sim		Rua do endereço do Leitor
<b>Número</b>	Varchar,10	Sim		Número da residência do Leitor

<b>Bairro</b>	Varchar,50	Sim		Bairro onde reside o leitor
<b>Cidade</b>	Varchar,50	Sim		Cidade onde reside o Leitor
<b>Cep</b>	Char,8	Sim		Cep da rua do Leitor
<b>Uf</b>	Char,2	Sim		Será inserido as siglas de cada estado,
<b>Id_login</b>	Integer	Sim	FK	Referência à tabela login

**Tabela 7.** emprestimo\_livro

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id_emprestimo</b>	Integer	Sim	PK	Referencia à Tabela Empréstimo, chave primária composta,
<b>Id_livro</b>	Integer	Sim	PK	Referencia à Tabela Livro. Chave primária composta,

**Tabela 8.** Editora

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id_editora</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Nome</b>	Varchar,100	Sim		Nome da editora
<b>Id_login</b>	Integer	Sim	FK	Referência à tabela login

## 2.5. Modelo Físico

### 2.5.1. *Scripts* DDL do banco de dados.

#### Criar Tabela Login

```
CREATE TABLE `login` (  
  `id_login` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `senha` VARCHAR(50) NOT NULL,  
  `tipo` VARCHAR(11) NOT NULL,  
  PRIMARY KEY (`id_login`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB;
```

#### Criar Tabela Autor

```
CREATE TABLE `autor` (  
  `id_autor` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `id_login` INT NOT NULL,  
  PRIMARY KEY (`id_autor`),  
  INDEX `FK_autor_login` (`id_login`),  
  CONSTRAINT `FK_autor_login` FOREIGN KEY (`id_login`) REFERENCES  
  `login` (`id_login`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB;
```

#### Criar Tabela Editora

```
CREATE TABLE `editora` (  
  `id_editora` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `id_login` INT NOT NULL,  
  PRIMARY KEY (`id_editora`),  
  INDEX `FK_editora_login` (`id_login`),  
  CONSTRAINT `FK_editora_login` FOREIGN KEY (`id_login`) REFERENCES  
  `login` (`id_login`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB;
```

#### Criar Tabela Livro

```
CREATE TABLE `livro` (  
  `id_livro` INT NOT NULL AUTO_INCREMENT,  
  `titulo` VARCHAR(100) NOT NULL,  
  `volume` INT NOT NULL,
```

```

`ano` CHAR(4) NOT NULL,
`id_login` INT NOT NULL,
`id_editora` INT NOT NULL,
`id_autor` INT NOT NULL,
PRIMARY KEY (`id_livro`),
INDEX `FK_livro_login` (`id_login`),
INDEX `FK_livro_editora` (`id_editora`),
INDEX `FK_livro_autor` (`id_autor`),
CONSTRAINT `FK_livro_autor` FOREIGN KEY (`id_autor`) REFERENCES
`autor` (`id_autor`),
CONSTRAINT `FK_livro_editora` FOREIGN KEY (`id_editora`) REFERENCES
`editora` (`id_editora`),
CONSTRAINT `FK_livro_login` FOREIGN KEY (`id_login`) REFERENCES `login`
(`id_login`)
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB;

```

### Criar Tabela Funcionário

```

CREATE TABLE `funcionario` (
`id_funcionario` INT NOT NULL AUTO_INCREMENT,
`nome` VARCHAR(100) NOT NULL,
`senha` CHAR(50) NOT NULL,
`id_login` INT NOT NULL,
PRIMARY KEY (`id_funcionario`),
INDEX `FK_funcionario_login` (`id_login`),
CONSTRAINT `FK_funcionario_login` FOREIGN KEY (`id_login`) REFERENCES
`login` (`id_login`)
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
AUTO_INCREMENT=21;

```

### Criar Tabela Leitor

```

CREATE TABLE `leitor` (
`id_leitor` INT NOT NULL AUTO_INCREMENT,
`nome` VARCHAR(100) NOT NULL,
`cpf` CHAR(11) NOT NULL,
`data_nascimento` DATE NOT NULL,
`telefone` CHAR(13) NOT NULL,
`email` VARCHAR(100) NOT NULL,
`rua` VARCHAR(100) NOT NULL,
`numero` VARCHAR(10) NOT NULL,
`bairro` VARCHAR(50) NOT NULL,
`cidade` VARCHAR(50) NOT NULL,
`cep` CHAR(8) NOT NULL,
`uf` CHAR(2) NOT NULL,
`id_login` INT NOT NULL,
PRIMARY KEY (`id_leitor`),
INDEX `FK_leitor_login` (`id_login`),
CONSTRAINT `FK_leitor_login` FOREIGN KEY (`id_login`) REFERENCES
`login` (`id_login`)
)

```

```
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB;
```

### Criar Tabela Empréstimo

```
CREATE TABLE `emprestimo` (  
  `id_emprestimo` INT NOT NULL AUTO_INCREMENT,  
  `data_emprestimo` DATE NOT NULL,  
  `data_devolucao` DATE NOT NULL,  
  `id_funcionario` INT NOT NULL,  
  `id_leitor` INT NOT NULL,  
  PRIMARY KEY (`id_emprestimo`),  
  INDEX `id_funcionario` (`id_funcionario`),  
  INDEX `id_leitor` (`id_leitor`),  
  CONSTRAINT `FK_emprestimo_funcionario` FOREIGN KEY (`id_funcionario`) REFERENCES `funcionario` (`id_funcionario`),  
  CONSTRAINT `FK_emprestimo_leitor` FOREIGN KEY (`id_leitor`) REFERENCES `leitor` (`id_leitor`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB;
```

### Criar Tabela Empréstimo\_Livro

```
CREATE TABLE `emprestimo_livro` (  
  `id_emprestimo` INT NOT NULL AUTO_INCREMENT,  
  `id_livro` INT NOT NULL,  
  INDEX `FK_emprestimo_livro_emprestimo` (`id_emprestimo`),  
  INDEX `FK_emprestimo_livro_livro` (`id_livro`),  
  CONSTRAINT `FK_emprestimo_livro_emprestimo` FOREIGN KEY (`id_emprestimo`) REFERENCES `emprestimo` (`id_emprestimo`),  
  CONSTRAINT `FK_emprestimo_livro_livro` FOREIGN KEY (`id_livro`) REFERENCES `livro` (`id_livro`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB;
```

#### 2.5.2. Comandos *Insert*

Visando alimentar cada tabela com 5 registros.

```
insert into login (nome, senha, tipo) values ('Otávio','tico3339','Funcionario');  
insert into login (nome, senha, tipo) values ('Ana Luiza','admin789','Admin');  
insert into login (nome, senha, tipo) values ('Dory','abc321','Funcionario');  
insert into login (nome, senha, tipo) values ('Rosi','dory123','Funcionario');  
insert into login (nome, senha, tipo) values ('Dell','123123','Funcionario');  
insert into login (nome, senha, tipo) values ('Jefferson','del4321','Funcionario');  
  
insert into autor (nome, id_login) values ('Susan Cain',2);  
insert into autor (nome, id_login) values ('Roger S. Pressman',2);  
insert into autor (nome, id_login) values ('Viktor Mayer-Schonberger',2);  
insert into autor (nome, id_login) values ('Paul Barry',2);
```

```

insert into autor (nome, id_login) values ('Richard Hunter',2);

insert into editora (nome, id_login) values ('Sextante',2);
insert into editora (nome, id_login) values ('AMGH',2);
insert into editora (nome, id_login) values ('Elsevier',2);
insert into editora (nome, id_login) values ('Alta Books',2);
insert into editora (nome, id_login) values ('MBOOKS',2);

insert into funcionario (nome, senha, id_login) values ('Jefferson','acb123',2);
insert into funcionario (nome, senha, id_login) values ('Ana','123abc',2);
insert into funcionario (nome, senha, id_login) values ('Dory','dory123',2);
insert into funcionario (nome, senha, id_login) values ('Rose','rosi4321',2);
insert into funcionario (nome, senha, id_login) values ('Dell','Dell789',2);

insert into leitor (nome, cpf, data_nascimento, telefone, email, rua, numero,
bairro, cidade, cep, uf, id_login) VALUES ('Miguel','47957368042','1970-11-
06','(71)999910300','miguel@gmail.com','Avenida Rosa','402','Alto das
Pombas','Recife','40226175','BA',2);
insert into leitor (nome, cpf, data_nascimento, telefone, email, rua, numero,
bairro, cidade, cep, uf, id_login) VALUES ('Arthur','68166208016','1959-02-
08','(92)999983558','Arthur@gmail.com','Rua
Independência','690','Alvorada','Olinda','69043020','AM',2);
insert into leitor (nome, cpf, data_nascimento, telefone, email, rua, numero,
bairro, cidade, cep, uf, id_login) VALUES ('Davi','96790166092','1961-10-
08','(41)999917759','Davi@gmail.com','Rua Hércules
Accorsi','860','Cafezal','Recife','86045270','PR',2);
insert into leitor (nome, cpf, data_nascimento, telefone, email, rua, numero,
bairro, cidade, cep, uf, id_login) VALUES ('Bernardo','50224973010','1965-09-
20','(62)999368617','Bernardo@gmail.com','Rua 86A','740','Setor
Sul','Cabo','74083340','GO',2);
insert into leitor (nome, cpf, data_nascimento, telefone, email, rua, numero,
bairro, cidade, cep, uf, id_login) VALUES ('Murilo','77913644095','1966-10-
29','(21)999366299','Murilo@gmail.com','Rua Projetada
A','230','Guaratiba','Camaragibe','23031195','RJ',2);

insert into livro (titulo, volume, ano, id_login, id_editora, id_autor) values ('Redes
de Computadores',2,'2011',2,1,5);
insert into livro (titulo, volume, ano, id_login, id_editora, id_autor) values
('Criptografia Para Iniciantes',1,'2012',2,2,4);
insert into livro (titulo, volume, ano, id_login, id_editora, id_autor) values ('Use a
Cabeça! Python',1,'2018',2,3,3);
insert into livro (titulo, volume, ano, id_login, id_editora, id_autor) values ('Use a
Cabeça!: Java ',1,'2007',2,4,3);
insert into livro (titulo, volume, ano, id_login, id_editora, id_autor) values ('Java
Para Leigos',1,'2013',2,5,1);

insert into emprestimo (data_emprestimo, data_devolucao, id_funcionario,
id_leitor) VALUES ("2019-08-01","2019-08-08",25,5);
insert into emprestimo (data_emprestimo, data_devolucao, id_funcionario,
id_leitor) VALUES ("2019-08-03","2019-08-10",25,4);
insert into emprestimo (data_emprestimo, data_devolucao, id_funcionario,
id_leitor) VALUES ("2019-08-12","2019-08-19",21,3);
insert into emprestimo (data_emprestimo, data_devolucao, id_funcionario,
id_leitor) VALUES ("2020-04-01","2020-04-08",22,2);
insert into emprestimo (data_emprestimo, data_devolucao, id_funcionario,
id_leitor) VALUES ("2020-04-02","2020-04-09",24,1);

```

```

insert into emprestimo_livro (id_emprestimo, id_livro) values (1,5);
insert into emprestimo_livro (id_emprestimo, id_livro) values (2,4);
insert into emprestimo_livro (id_emprestimo, id_livro) values (3,3);
insert into emprestimo_livro (id_emprestimo, id_livro) values (4,1);
insert into emprestimo_livro (id_emprestimo, id_livro) values (5,2);

```

### 2.5.3. Relatórios

Os relatórios que foram solicitados com os respectivos comandos *Select*.

1 - Listar todos os dados dos livros cadastrados, ordenando pelo seu título.

```

SELECT * FROM livro
ORDER BY titulo

```

2 - Listar o funcionário de código 25 e o nome do administrador que o cadastrou.

```

SELECT      funcionario.id_funcionario,      funcionario.nome      AS
Nome_Funcionario, login.nome AS Nome_admin, login.tipo
FROM funcionario, login
WHERE login.tipo = 'admin' AND funcionario.id_funcionario = 25

```

3 - Informar a quantidade de empréstimos feitos por nome de funcionários, realizados no mês de agosto de 2019.

```

SELECT f.nome AS Nome_Funcionario, COUNT(*) Quant_Emprestimo
FROM emprestimo e, funcionario f
WHERE e.id_funcionario = f.id_funcionario
AND e.data_emprestimo BETWEEN '2019-08-01' AND '2019-08-31'
GROUP BY f.nome

```

4 - Listar as quantidades de livros por autor (nome).

```

SELECT nome AS Nome_autor, COUNT(*) Quant_Livro
FROM autor, livro
WHERE livro.id_autor = autor.id_autor
GROUP BY autor.nome

```

5 - Listar os nomes das editoras em ordem crescente.

```

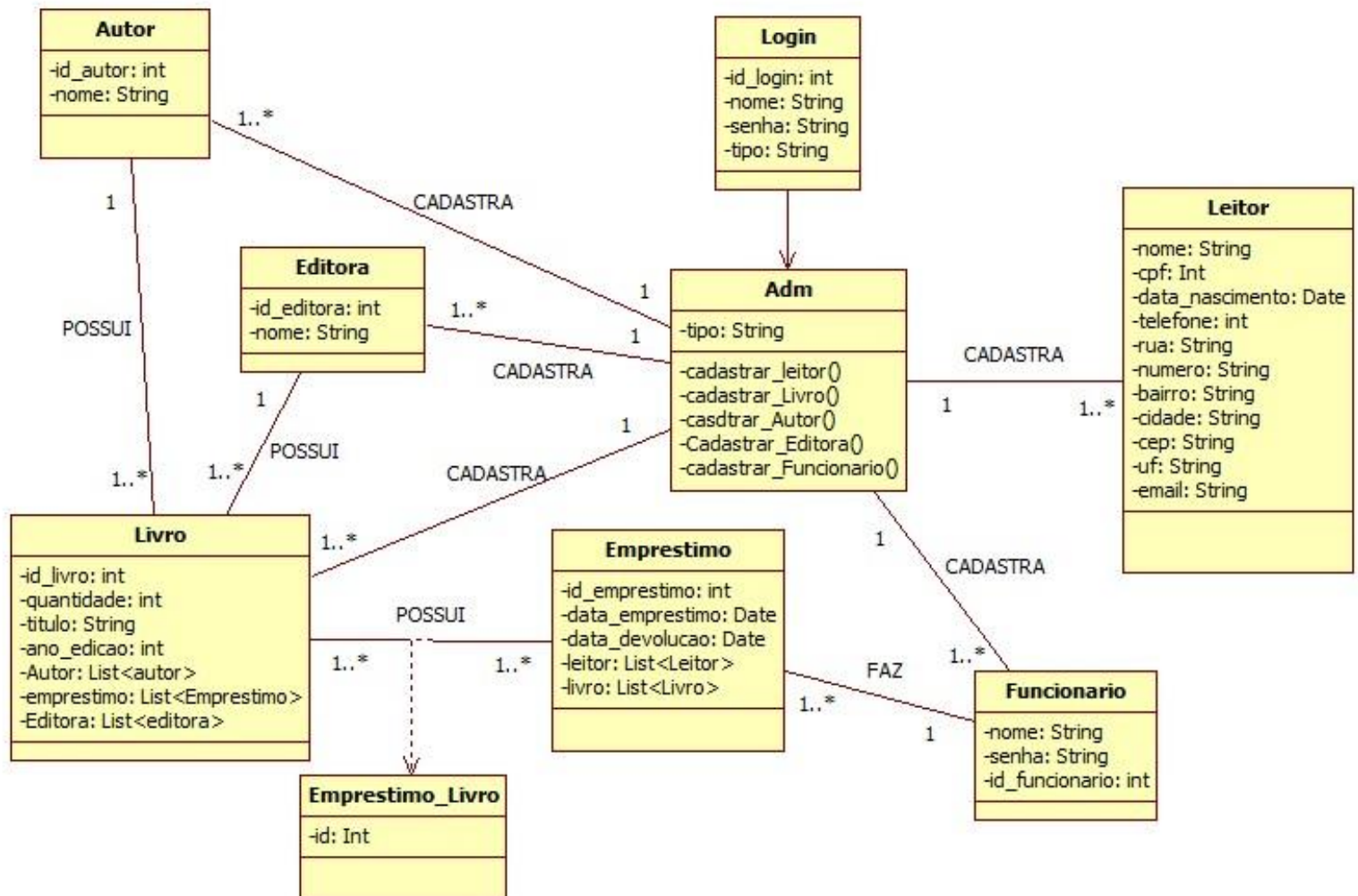
SELECT nome as Nome_editora_Ordem_Crescente FROM editora
ORDER BY nome

```

## 3. Análise e Design

### 3.1. Diagrama de Classes

Ferramenta utilizada: StarUML





## 4. Arquitetura do Software

### 4.1. Padrão de projeto

Deverá ser apresentada uma breve descrição sobre o padrão MVC e descrever as classes que serão implementadas em cada uma das camadas.

### 4.2. Protótipo

Deverão ser apresentadas cada uma das telas do sistema para cada uma das camadas.

Para cada tela deverão ser descritos os requisitos funcionais que a mesma atende.