



**UNINABUCO**

ANÁLISE E DESENVOLVIMENTO DE SOFTWARE

TÓPICOS INTEGRADORES II

# Documentação de Projeto de Software

## Versão 2.0

Autor(a): Jefferson Queiroz Pereira de Melo

[Link do Github](#)

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.

# 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 .....	4
2.1. Modelo de Visão .....	4
2.2. Modelo Conceitual.....	6
2.3. Modelo Lógico .....	7
2.4. Dicionário de Dados .....	8
2.5. Modelo Físico .....	11
2.5.1. Deverão ser apresentado os scripts DDL do banco de dados...	11
2.5.2. Comandos Insert para alimentar cada tabela com 5 registros.	14
2.5.3. Os relatórios que foram solicitados e os respectivos comandos Select.	15
3. Análise e Design .....	15
3.1. Diagrama de Classes .....	15
4. Arquitetura do Software.....	16
4.1. Padrão de projeto .....	16
4.2. Protótipo .....	16

# 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

Este é um banco de dados de controle de biblioteca, cujo programa oferece opções de cadastrar, alterar, consultar e excluir leitores e acervos. Todo o processo será feito por um único usuário que é o administrador do sistema.

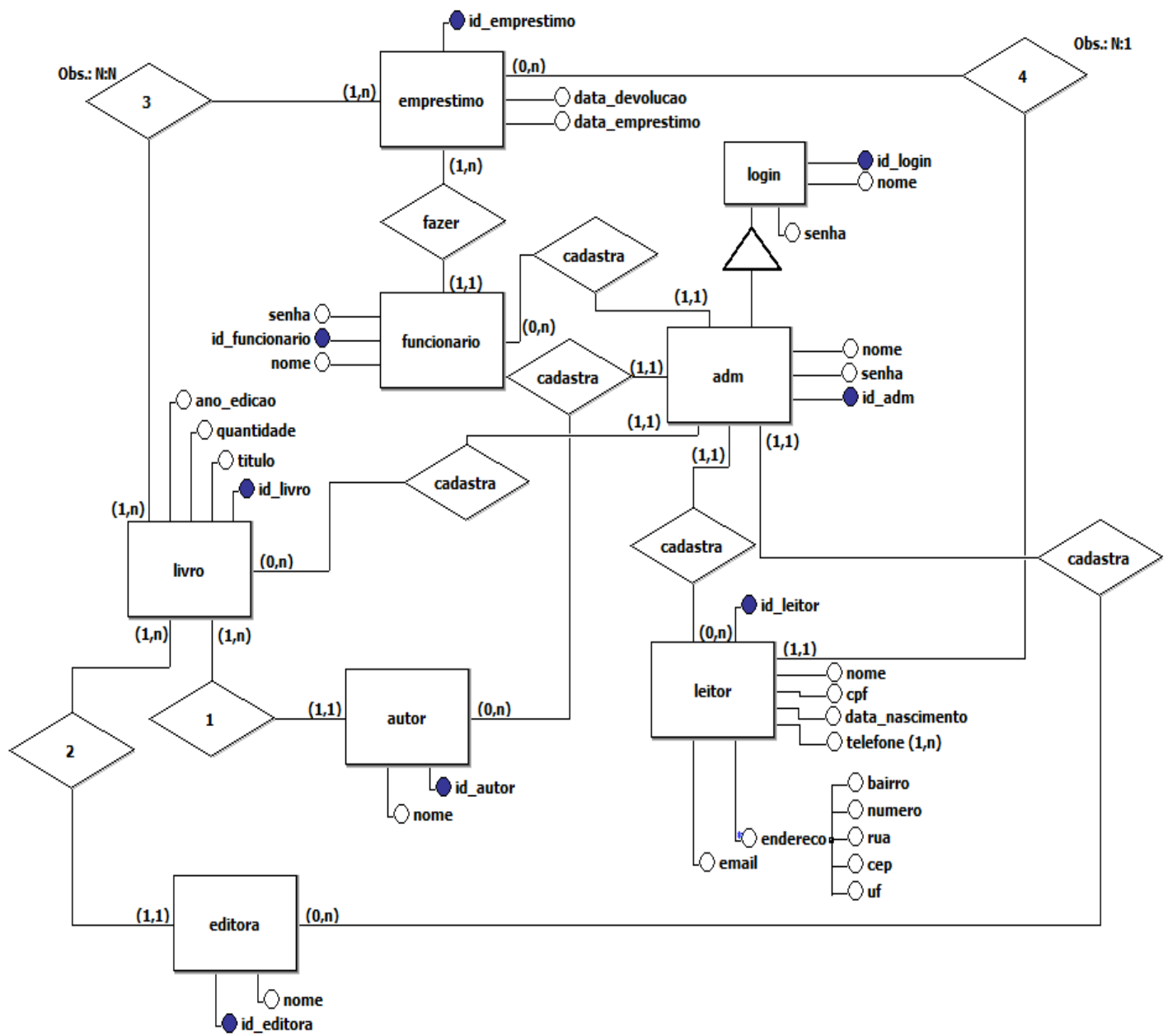
- O primeiro login deve ser feito pelo administrador, só ele poderá cadastrar leitores e acervos.

- 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*.
- No cadastro dos acervos, será solicitado: título da obra, autor, editora, classificação literária, ano de edição, número de páginas, quantidade. Também haverá um ID, o *Primary Key*.
- No empréstimo, será requisitado o nome do leitor previamente cadastrado. Este escolherá um ou mais acervo e será atribuído um ID, o *Primary Key*.
- Na devolução, pode ser requerido o ID do empréstimo ou o nome do leitor. O prazo para a 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 programa possibilitará a extração de relatórios de várias formas:

- Relação dos leitores;
- Relação dos acervos;
- Quantidade de cervos 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)**

id\_login – Primary Key

**adm (id\_adm, nome, senha)**

id\_adm – Primary Key

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

id-autor – Primary Key

id\_amd referencia adm - Foreign Key

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

id\_editora - Primary Key

id\_adm referencia adm - Foreign Key

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

id\_livro - Primary Key

id\_adm referencia adm - Foreign Key

id\_editora referencia editora - Foreign Key

Id\_autor referencia autor - Foreign Key

**leitor (id\_leitor, nome, cpf, data\_nascimento, telefone, endereco, email, id\_adm)**

id\_leitor - Primary Key

id\_adm referencia adm - Foreign Key

**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\_adm)**

id\_funcionario – Primary Key

id\_adm referencia adm - Foreign Key

**emprestimo livro (cod\_emprestimo, cod\_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

**Tabela 2. Adm**

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

**Tabela 3. 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_adm</b>	Integer	Sim	FK	Referência à Tabela Adm

**Tabela 4. 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_edição</b>	Integer	Sim		Ano da edição do livro
<b>Id_adm</b>	Integer	Sim	FK	Referencia à Tabela adm
<b>Id_editora</b>	Integer	Sim	FK	Referencia à Tabela Editora
<b>Id_autor</b>	Integer	Sim	FK	Referencia à Tabela Autor

**Tabela 5.** Funcionario

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Nome</b>	Varchar,10	Sim		Nome do funcionário,
<b>Senha</b>	Varchar,50	Sim		Senha do funcionário
<b>Id_adm</b>	Varchar,5	Sim	FK	Referencia à tabela Adm

**Tabela 6.** Emprestimo

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
<b>Id</b>	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
<b>Data_emprestimo</b>	Varchar,10	Sim		Data da locação do(s) Livro(s)
<b>Data_devolucao</b>	Varchar,10	Sim		Data da entrega do(s) Livro(s)
<b>Id_funcionario</b>	Varchar	Sim	FK	Referencia à Tabela Funcionario
<b>Id_leitor</b>	Integer	Sim	FK	Referencia à Tabela Leitor

**Tabela 7.** 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>	Varchar,11	Sim		Cpf do leitor
<b>Data_nascimento</b>	Varchar,10	Sim		Data de nascimento do Leitor

<b>Telefone</b>	Varchar,15	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>Numero</b>	Varchar,10	Sim		Número da residência do Leitor
<b>Cidade</b>	Varchar,50	Sim		Cidade onde reside o leitor
<b>Bairro</b>	Varchar,50	Sim		Bairro onde reside o Leitor
<b>Cep</b>	Varchar,10	Sim		Cep da rua do Leitor
<b>Uf</b>	Varchar,2	Sim		Será inserido as siglas de cada estado,
<b>Id_adm</b>	Integer	Sim	FK	Referência à tabela Adm

**Tabela 8.** 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 9.** 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_adm</b>	Integer	Sim	FK	Referência à tabela Adm

## 2.5. Modelo Físico

2.5.1. Deverão ser apresentado os scripts DDL do banco de dados.

### Criar Tabela Login

```
CREATE TABLE `login` (  
  `id_login` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL COLLATE 'utf8mb4_general_ci',  
  `senha` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_general_ci',  
  PRIMARY KEY (`id_login`)  
)  
COLLATE='utf8mb4_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=6  
;
```

### Criar Tabela Adm

```
CREATE TABLE `adm` (  
  `id_adm` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL COLLATE 'utf8mb4_general_ci',  
  `senha` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_general_ci',  
  PRIMARY KEY (`id_adm`)  
)  
COLLATE='utf8mb4_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=207  
;
```

### Criar Tabela Autor

```
CREATE TABLE `autor` (  
  `id_autor` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `id_adm` INT NOT NULL,  
  PRIMARY KEY (`id_autor`),  
  INDEX `FK_autor_adm` (`id_adm`),  
  CONSTRAINT `FK_autor_adm` FOREIGN KEY (`id_adm`) REFERENCES  
  `adm` (`id_adm`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=6  
;
```

### Criar Tabela Editora

```
CREATE TABLE `editora` (  
  `id_editora` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `id_adm` INT NOT NULL,  
  PRIMARY KEY (`id_editora`),  
  INDEX `FK_editora_adm` (`id_adm`),  
  CONSTRAINT `FK_editora_adm` FOREIGN KEY (`id_adm`)   
  REFERENCES `adm` (`id_adm`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=6;
```

### Criar Tabela Livro

```
CREATE TABLE `livro` (  
  `id_livro` INT NOT NULL AUTO_INCREMENT,  
  `titulo` VARCHAR(100) NOT NULL,  
  `volume` INT NOT NULL,  
  `ano` INT NOT NULL,  
  `id_adm` INT NOT NULL,  
  `id_editora` INT NOT NULL,  
  `id_autor` INT NOT NULL,  
  PRIMARY KEY (`id_livro`),  
  INDEX `FK_livro_adm` (`id_adm`),  
  INDEX `FK_livro_editora` (`id_editora`),  
  INDEX `FK_livro_autor` (`id_autor`),  
  CONSTRAINT `FK_livro_adm` FOREIGN KEY (`id_adm`) REFERENCES   
  `adm` (`id_adm`),  
  CONSTRAINT `FK_livro_autor` FOREIGN KEY (`id_autor`)   
  REFERENCES `autor` (`id_autor`),  
  CONSTRAINT `FK_livro_editora` FOREIGN KEY (`id_editora`)   
  REFERENCES `editora` (`id_editora`)  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=6  
;
```

### Criar Tabela Funcionário

```
CREATE TABLE `funcionario` (  
  `id_funcionario` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `senha` VARCHAR(50) NOT NULL,  
  `id_adm` INT NOT NULL,  
  PRIMARY KEY (`id_funcionario`),  
  INDEX `FK_funcionario_adm` (`id_adm`),  
  CONSTRAINT `FK_funcionario_adm` FOREIGN KEY (`id_adm`)   
  REFERENCES `adm` (`id_adm`)
```

```
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
AUTO_INCREMENT=5
;
```

### Criar Tabela Leitor

```
CREATE TABLE `leitor` (
  `id_leitor` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(100) NOT NULL,
  `cpf` VARCHAR(11) NOT NULL,
  `data_nascimento` VARCHAR(10) NOT NULL,
  `telefone` VARCHAR(15) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `rua` VARCHAR(100) NOT NULL,
  `numero` VARCHAR(10) NOT NULL,
  `cidade` VARCHAR(50) NOT NULL,
  `bairro` VARCHAR(50) NOT NULL,
  `cep` VARCHAR(10) NOT NULL,
  `uf` VARCHAR(2) NOT NULL,
  `id_adm` INT NOT NULL,
  PRIMARY KEY (`id_leitor`),
  INDEX `FK_leitor_adm` (`id_adm`),
  CONSTRAINT `FK_leitor_adm` FOREIGN KEY (`id_adm`) REFERENCES
  `adm` (`id_adm`)
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
AUTO_INCREMENT=6
;
```

### Criar Tabela Empréstimo

```
CREATE TABLE `emprestimo` (
  `id_emprestimo` INT NOT NULL AUTO_INCREMENT,
  `data_emprestimo` VARCHAR(10) NOT NULL,
  `data_devolucao` VARCHAR(10) NOT NULL,
  `id_funcionario` INT NOT NULL,
  `id_leitor` INT NOT NULL,
  PRIMARY KEY (`id_emprestimo`),
  INDEX `FK_emprestimo_funcionario` (`id_funcionario`),
  INDEX `FK_emprestimo_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
AUTO_INCREMENT=6
;
```

### Criar Tabela Empréstimo\_Livro

```
CREATE TABLE `emprestimo_livro` (  
  `id_emprestimo` INT NOT NULL,  
  `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 para alimentar cada tabela com 5 registros.

```
insert into adm values (1,'administrador',1234);
```

```
insert into login values (1,'Jefferson','abc123');
```

```
insert into login values (2,'Ana','abc321');
```

```
insert into login values (3,'Dory','dory123');
```

```
insert into login values (4,'Rosi','123123');
```

```
insert into login values (5,'dell','del4321');
```

```
insert into autor values (1,'Susan Cain',1);
```

```
insert into autor values (2,'Roger S. Pressman ',1);
```

```
insert into autor values (3,'Viktor Mayer-Schonberger ',1);
```

```
insert into autor values (4,'Paul Barry ',1);
```

```
insert into autor values (5,'Richard Hunter',1);
```

```
insert into editora values (1,'Sextante',1);
```

```
insert into editora values (2,'AMGH',1);
```

```
insert into editora values (3,'Elsevier',1);
```

```
insert into editora values (4,'Alta Books',1);
```

```
insert into editora values (5,'MBOOKS',1);
```

```
insert into funcionario values (1,'Jefferson','acb123',1);
```

```
insert into funcionario values (2,'Ana','123abc',1);
```

```
insert into funcionario values (3,'Dory','dory123',1);
```

```
insert into funcionario values (4,'Rose','rosi4321',1);
```

```
insert into funcionario values (5,'Dell','Dell789',1);
```

```

insert into leitor
values(1,'Miguel','47957368042','06/11/1970','(71)999910300','Miguel@gmail.com','Avenida Rosa','402','Recife','Alto das Pombas','40226-175','BA',1);

insert into leitor
values(2,'Arthur','68166208016','08/02/1959','(92)999983558','Arthur@gmail.com','Rua Independência','690','Olinda','Alvorada','69043-020','AM',1);

insert into leitor
values(3,'Davi','96790166092','08/10/1961','(41)999917759','Davi@gmail.com','Rua Hércules Accorsi','860','Recife','Cafezal','86045-270','PR',1);

insert into leitor
values(4,'Bernardo','50224973010','20/09/1965','(62)999368617','Bernardo@gmail.com','Rua 86A','740','Cabo','Setor Sul','74083-340','GO',1);

insert into leitor
values(5,'Murilo','77913644095','20/10/1966','(21)999366299','Murilo@gmail.com','Rua Projetada A','230','Camaragibe','Guaratiba','23031-195','RJ',1);

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

insert into emprestimo values (1,'29/03/20','06/04/20',1,5);
insert into emprestimo values (2,'30/03/20','07/04/20',2,4);
insert into emprestimo values (3,'31/03/20','08/04/20',3,3);
insert into emprestimo values (4,'01/04/20','09/04/20',4,2);
insert into emprestimo values (5,'02/04/20','10/04/20',2,1);

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

```

2.5.3. Os relatórios que foram solicitados e os respectivos comandos Select.

## 3. Análise e Design

### 3.1. Diagrama de Classes

Será apresentado o diagrama de classes básicas e deverá ser informada qual ferramenta foi utilizada e apresentada a imagem final do diagrama.

## 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.