



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: <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: Livro_editora e Livro_Autor , acrescentado chaves estrangeira na tabela livro.

Sumário

1. Introdução ao Documento	5
1.1. Área de negócio do Sistema	5
1.2. Principais funcionalidades.....	5
1.3. Método de trabalho	5
2. Modelo de Dados	5
2.1. Modelo de Visão	5
2.2. Modelo Conceitual.....	7
2.3. Modelo Lógico	8
2.4. Dicionário de Dados	9
2.5. Modelo Físico	12
2.5.1. Deverão ser apresentado os scripts DDL do banco de dados...	12
2.5.2. Comandos Insert para alimentar cada tabela com 5 registros.	15
2.5.3. Os relatórios que foram solicitados e os respectivos comandos Select.	16
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

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.

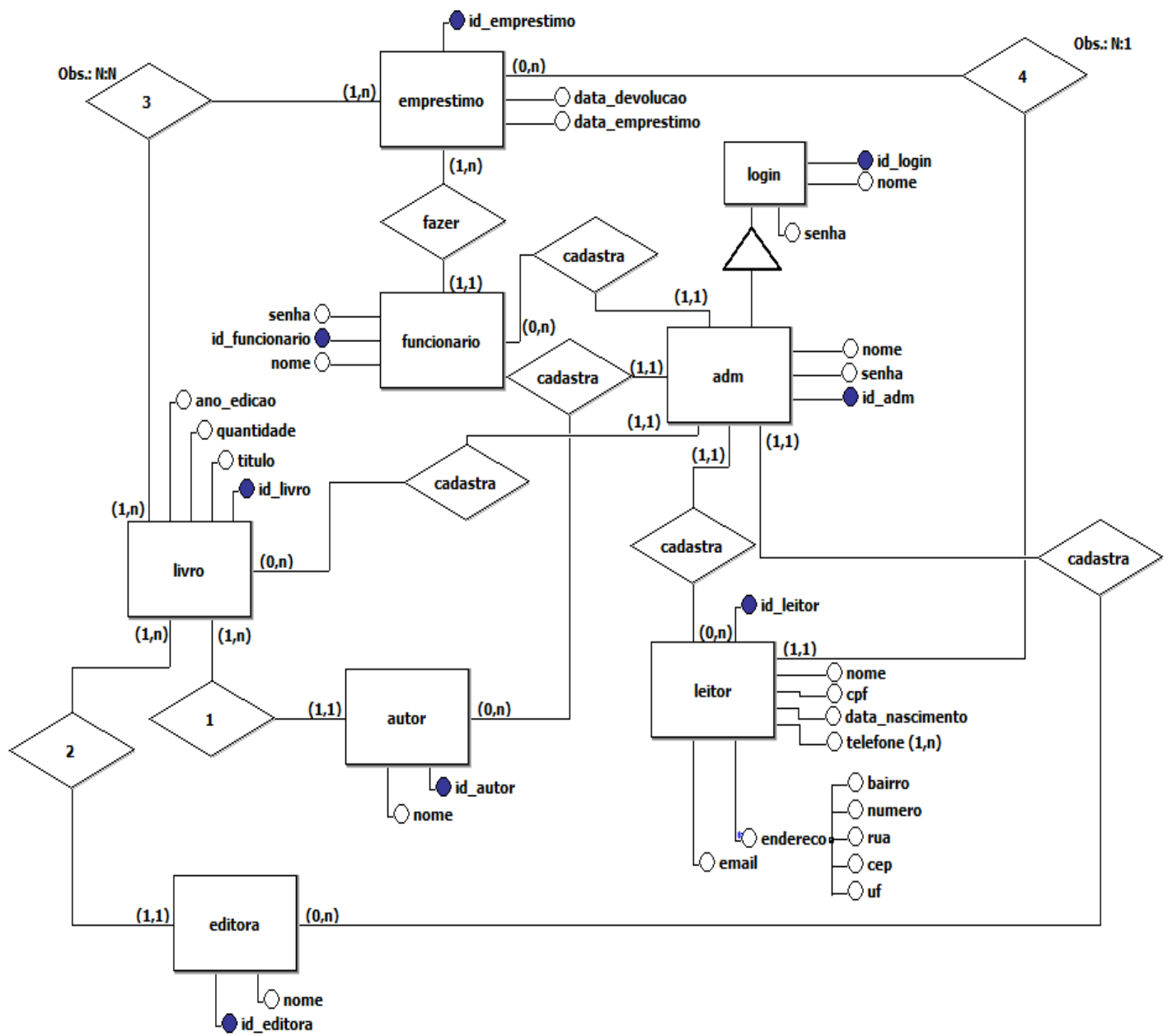
- O primeiro login deve ser feito pelo administrador, só ele poderá cadastrar leitores e acervos.

- 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
Id	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
Nome	Varchar,100	Sim		Nome do usuário que está logando
Senha	Varchar,50	Sim		Senha do usuário que está logando

Tabela 2. Adm

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
Id	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
Nome	Varchar,100	Sim		Nome do usuário que está logando
Senha	Varchar,50	Sim		Senha do usuário que está logando

Tabela 3. Autor

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
Id	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
Nome	Varchar,100	Sim		Nome do autor
Id_adm	Integer	Sim	FK	Referência à Tabela Adm

Tabela 4. Livro

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
Id	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment

Título	Varchar,100	Sim		Nome do Livro,
Volume	Integer	Sim		Quantidade em estoque
Ano_edição	Integer	Sim		Ano da edição do livro
Id_adm	Integer	Sim	FK	Referencia à Tabela adm
Id_editora	Integer	Sim	FK	Referencia à Tabela Editora
Id_autor	Integer	Sim	FK	Referencia à Tabela Autor

Tabela 5. Funcionario

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

Tabela 6. Emprestimo

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

Tabela 7. Leitor

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
Id	Integer	Sim	PK	Chave Primária da tabela, Auto_Increment
Nome	Varchar,100	Sim		Nome do Leitor
Cpf	Varchar,11	Sim		Cpf do leitor
Data_nascimento	Varchar,10	Sim		Data de nascimento do Leitor

Telefone	Varchar,15	Sim		Telefone do Leitor
Email	Varchar,100	Sim		E-mail do Leitor
Rua	Varchar,100	Sim		Rua do endereço do Leitor
Numero	Varchar,10	Sim		Número da residência do Leitor
Cidade	Varchar,50	Sim		Cidade onde reside o leitor
Bairro	Varchar,50	Sim		Bairro onde reside o Leitor
Cep	Varchar,10	Sim		Cep da rua do Leitor
Uf	Varchar,2	Sim		Será inserido as siglas de cada estado,
Id_adm	Integer	Sim	FK	Referência à tabela Adm

Tabela 8. emprestimo_livro

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

Tabela 9. Editora

Identificador	Tipo (tamanho)	Obrigatório	Chave	Descrição
Id_editora	Integer	Sim	PK	Chave Primária da tabela, Auto Increment
nome	Varchar,100	Sim		Nome da editora
Id_adm	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.