



**Campus Campina Grande**

**Cinema em Dados: Modelagem Relacional e Especificação de um Banco de Dados para  
Gerenciamento de Informações do Cinema**

**GISLANY DIAS FORMIGA  
ICARO MENDES DE ALCANTARA  
LÍVIA MARIA FORTUNATO DE SOUSA**

Área do conhecimento: Ciência da Computação  
Área Temática: Tecnologia da Informação e Comunicação

**CAMPINA GRANDE**

**2023**

**2023**

**ICARO MENDES DE ALCANTARA BEZERRA**

**GISLANY DIAS FORMIGA**

**LÍVIA MARIA FORTUNATO DE SOUSA**

**Cinema em Dados: Modelagem Relacional e Especificação de um Banco de  
Dados para Gerenciamento de Informações do Cinema**

Projeto avaliativo da disciplina de  
Banco de Dados lecionada pelo  
Professor Danyllo Albuquerque,  
responsável pela referida disciplina  
componente do curso superior em  
Engenharia da Computação.

**CAMPINA GRANDE**

**2023**

## RESUMO

A criação de uma modelagem relacional e especificação de um banco de dados para gerenciamento de informações do cinema é uma prática relevante e necessária para aprimorar as operações do cinema, aumentar a satisfação dos clientes e garantir a competitividade no mercado. Em primeiro lugar, o uso de um banco de dados bem projetado permite o armazenamento organizado e acessível de uma grande quantidade de informações relacionadas às operações do cinema, como dados dos filmes, horários de exibição, vendas de ingressos, informações dos clientes e dados financeiros.

Ademais, a implementação de um banco de dados eficiente contribui para a melhoria da eficiência operacional do cinema, ao permitir a automação de processos e a geração de relatórios gerenciais que auxiliam na tomada de decisões. Por sua vez, a otimização das operações proporciona benefícios tanto para o cinema quanto para os clientes, por meio da redução de custos, do aumento da produtividade e do aumento da qualidade dos serviços prestados.

Outro aspecto a ser destacado é a melhoria na experiência do usuário. Um banco de dados bem projetado permite a implementação de sistemas de reservas online, programas de fidelidade, dentre outras funcionalidades que podem aumentar a satisfação dos clientes e fidelizá-los ao cinema.

Assim, a criação de uma modelagem relacional e especificação de um banco de dados para gerenciamento de informações do cinema é uma prática que apresenta benefícios significativos para o sucesso do negócio e a satisfação dos clientes. É fundamental que o banco de dados seja projetado de forma cuidadosa e criteriosa, levando em consideração as particularidades do cinema e os objetivos da empresa. Dessa forma, será possível maximizar os benefícios obtidos a partir da implementação de um banco de dados eficiente e satisfazer as necessidades do cinema e de seus clientes.

## **LISTA DE FIGURAS**

<b>FIGURA 1 - DIAGRAMA ENTIDADE RELACIONAMENTO</b>	<b>9</b>
<b>FIGURA 2 - AUTO RELACIONAMENTO DE FUNCIONÁRIO</b>	<b>10</b>
<b>FIGURA 3 - ESPECIFICAÇÃO DE FUNCIONÁRIO</b>	<b>11</b>
<b>FIGURA 4 - ESPECIFICAÇÃO DE CLIENTE</b>	<b>12</b>
<b>FIGURA 5 - MODELO LÓGICO</b>	<b>13</b>
<b>FIGURA 6 - TABELA DE CADASTRO DE FUNCIONÁRIOS DO CINEMA</b>	<b>14</b>
<b>FIGURA 7 - TABELA DE CADASTRO DE INFORMAÇÕES DO CINEMA</b>	<b>14</b>
<b>FIGURA 8 - TABELA DE CADASTRO DE INFORMAÇÕES DOS FILMES</b>	<b>15</b>
<b>FIGURA 9 - TABELA DE CADASTRO DE INFORMAÇÕES DOS CLIENTES</b>	<b>15</b>
<b>FIGURA 10 - TABELA DE CADASTRO DE INFORMAÇÕES DO BILHETE</b>	<b>15</b>
<b>FIGURA 11 - TABELA DE RESTRIÇÕES DA TABELA FUNCIONÁRIO</b>	<b>16</b>
<b>FIGURA 12 - TABELA DE RESTRIÇÕES DA TABELA CINEMA</b>	<b>16</b>
<b>FIGURA 13 - TABELA DE RESTRIÇÕES DA TABELA FILME</b>	<b>16</b>
<b>FIGURA 14 - TABELA DE RESTRIÇÕES DA TABELA CLIENTE</b>	<b>16</b>
<b>FIGURA 15 - TABELA DE RESTRIÇÕES DA TABELA BILHETE</b>	<b>17</b>
<b>FIGURA 16 - APLICAÇÃO DE DESCONTO DA MEIA ENTRADA</b>	<b>23</b>
<b>FIGURA 17 - CONSULTA SIMPLES</b>	<b>24</b>
<b>FIGURA 18 - CONSULTA COM JUNÇÃO</b>	<b>24</b>
<b>FIGURA 19 - ALTERAÇÃO DE DADOS</b>	<b>25</b>
<b>FIGURA 20 - EXCLUSÃO DE DADOS</b>	<b>25</b>
<b>FIGURA 21 - CONSULTA SIMPLES ENVOLVENDO UMA ÚNICA TABELA</b>	<b>25</b>
<b>FIGURA 22 - CONSULTA SIMPLES ENVOLVENDO DUAS TABELAS</b>	<b>26</b>
<b>FIGURA 23 - CONSULTA SIMPLES ENVOLVENDO TRÊS TABELAS</b>	<b>26</b>
<b>FIGURA 24 - CONSULTA ENVOLVENDO AGRUPAMENTO</b>	<b>27</b>
<b>FIGURA 25 - CONSULTA ENVOLVENDO FUNÇÃO AGREGADA</b>	<b>27</b>

# SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>6</b>
<b>1.1 DESCRIÇÃO DA REALIDADE</b>	<b>6</b>
<b>1.2 FINALIDADE DO SISTEMA</b>	<b>7</b>
<b>1.3 ESCOPO DO PROJETO</b>	<b>7</b>
<b>2. MODELO ENTIDADE RELACIONAMENTO</b>	<b>7</b>
<b>2.1 MONTAGEM DO DER</b>	<b>7</b>
<b>2.2 DIAGRAMA ENTIDADE RELACIONAMENTO</b>	<b>9</b>
<b>3. MODELO LÓGICO</b>	<b>10</b>
<b>3.1 MONTAGEM DO DIAGRAMA LÓGICO</b>	<b>10</b>
<b>3.2 DIAGRAMA LÓGICO</b>	<b>13</b>
<b>4. DICIONÁRIO DE DADOS</b>	<b>14</b>
<b>4.1 TABELA FUNCIONÁRIO</b>	<b>14</b>
<b>4.2 TABELA CINEMA</b>	<b>14</b>
<b>4.3 TABELA FILME</b>	<b>15</b>
<b>4.4 TABELA CLIENTE</b>	<b>15</b>
<b>4.5 TABELA BILHETE</b>	<b>15</b>
<b>5. RESTRIÇÕES</b>	<b>16</b>
<b>5.1 TABELA RESTRIÇÕES DE FUNCIONÁRIO</b>	<b>16</b>
<b>5.2 TABELA RESTRIÇÕES DE CINEMA</b>	<b>16</b>
<b>5.3 TABELA RESTRIÇÕES DE FILME</b>	<b>16</b>
<b>5.4 TABELA RESTRIÇÕES DE CLIENTE</b>	<b>16</b>
<b>5.5 TABELA RESTRIÇÕES DE BILHETE</b>	<b>17</b>
<b>6. MODELO FÍSICO</b>	<b>17</b>
<b>6.1 CRIAÇÃO DO BANCO DE DADOS</b>	<b>17</b>
<b>6.2 SCRIPTS: CRIAÇÃO DE TABELAS, ATRIBUTOS E CONSTRAINTS</b>	<b>17</b>
<b>7. INSERÇÃO DE DADOS</b>	<b>21</b>
<b>7.1 INSERÇÃO DE DADOS</b>	<b>21</b>
<b>7.2 APLICAÇÃO DE DESCONTO DA MEIA ENTRADA</b>	<b>23</b>
<b>8. CONSULTAS</b>	<b>24</b>

<b>8.1 CONSULTA SIMPLES</b>	<b>24</b>
<b>8.2 CONSULTA COM JUNÇÃO</b>	<b>24</b>
<b>8.3 ALTERAÇÃO DE DADOS</b>	<b>24</b>
<b>8.4 EXCLUSÃO DE DADOS</b>	<b>25</b>
<b>8.5 CONSULTA SIMPLES ENVOLVENDO UMA ÚNICA TABELA</b>	<b>25</b>
<b>8.6 CONSULTA SIMPLES ENVOLVENDO DUAS TABELAS</b>	<b>26</b>
<b>8.7 CONSULTA SIMPLES ENVOLVENDO TRÊS TABELAS</b>	<b>26</b>
<b>8.8 CONSULTA ENVOLVENDO AGRUPAMENTO</b>	<b>26</b>
<b>8.9 CONSULTA ENVOLVENDO FUNÇÃO AGREGADA</b>	<b>27</b>

## **1. INTRODUÇÃO**

Compreendendo a importância da gestão eficiente de informações para o setor cinematográfico, esta seção tem como objetivo apresentar as condições e necessidades específicas relacionadas à modelagem de um banco de dados voltado ao gerenciamento de informações sobre cinemas. Serão analisados os principais casos de uso e requisitos do sistema, visando à elaboração de uma estrutura de armazenamento de dados eficiente e coesa.

### **1.1 DESCRIÇÃO DA REALIDADE**

O cinema é uma das formas de arte mais populares em todo o mundo e é uma importante indústria cultural. No entanto, a pandemia de COVID-19 teve um grande impacto na indústria cinematográfica, afetando as atividades dos cinemas em todo o mundo. Com o fechamento de muitos cinemas e o cancelamento de estreias de filmes, a indústria sofreu um grande impacto financeiro.

Além disso, a indústria cinematográfica enfrenta desafios no gerenciamento de informações, desde o controle de bilheteria, passando pela gestão de sessões e horários, até o gerenciamento de informações sobre filmes, salas, sessões e atores. Um banco de dados bem projetado pode ajudar a gerenciar eficientemente todas essas informações e garantir que os cinemas operem de forma eficiente, oferecendo aos espectadores uma experiência agradável. Com um banco de dados bem projetado, os cinemas podem gerenciar efetivamente informações como programação de sessões, informações sobre filmes em exibição, dados de bilheteria, histórico de transações, informações de clientes, entre outros. Além disso, é possível gerar relatórios que auxiliem na tomada de decisões estratégicas, como quais filmes são mais populares em determinados horários ou em determinadas regiões.

Outro benefício de um banco de dados bem projetado é a capacidade de otimizar a comunicação entre os cinemas e os distribuidores de filmes. Com informações precisas e atualizadas sobre a disponibilidade de salas de cinema, os distribuidores podem agendar as estreias dos filmes de forma mais eficiente.

Portanto, a criação de um banco de dados bem projetado para gerenciar informações de cinemas pode ser fundamental para a sobrevivência e prosperidade da indústria cinematográfica, principalmente em momentos de crise, como o atual momento de pandemia.

## **1.2 FINALIDADE DO SISTEMA**

Este trabalho tem como objetivo modelar os dados do sistema de banco de dados para um cinema, abrangendo as três principais fases da modelagem: conceitual, lógica e física. Serão definidas as tabelas necessárias para armazenar as informações relevantes, bem como as relações entre elas, utilizando chaves primárias e estrangeiras para garantir a integridade e coerência dos dados.

## **1.3 ESCOPO DO PROJETO**

Este projeto visa suprir as demandas dos especialistas de TI da Cinesercla, em particular dos projetistas e administradores de banco de dados.

A pesquisa foi desenvolvida com o propósito de satisfazer as exigências do setor de cinema, possibilitando a administração eficiente dos processos gerenciais e operacionais, em conformidade com as metas do sistema previamente delineadas.

É importante destacar que o objetivo deste projeto é criar um modelo de armazenamento de dados utilizando o padrão relacional e a linguagem SQL. Serão utilizadas três perspectivas para a modelagem: conceitual, lógica e física, utilizando ferramentas como BRModelo, MySQL Workbench e Microsoft SQL Server 2007.

## **2. MODELO ENTIDADE RELACIONAMENTO**

### **2.1 MONTAGEM DO DER**

Pretende-se apresentar o funcionamento e estruturação do banco de dados para um sistema de gestão de informações do setor cinematográfico. Será descrita a organização das principais tabelas que compõem o banco, suas relações e finalidades, com o objetivo de subsidiar a modelagem da camada de armazenamento de dados.

No modelo relacional apresentado, existem diversas tabelas que se relacionam entre si. A tabela de funcionários possui uma especialização não exclusiva, de técnico de exibição, que tem como diferencial dos demais o atributo ‘certificado’, diferenciando o salário dos demais cargos. Essa tabela possui um auto-relacionamento através da gerência, o que significa que um funcionário pode gerenciar outros funcionários.

Por conseguinte, a tabela de funcionários também se relaciona com a tabela de cinema, indicando que cada funcionário pode trabalhar em um cinema específico. Essa tabela de cinema, por sua vez, se relaciona com a tabela de salas, indicando qual cinema possui quais salas de exibição. A tabela de funcionários do tipo técnico de exibição também se relaciona com a tabela de salas, indicando quais salas e técnicos são responsáveis pela exibição dos filmes.

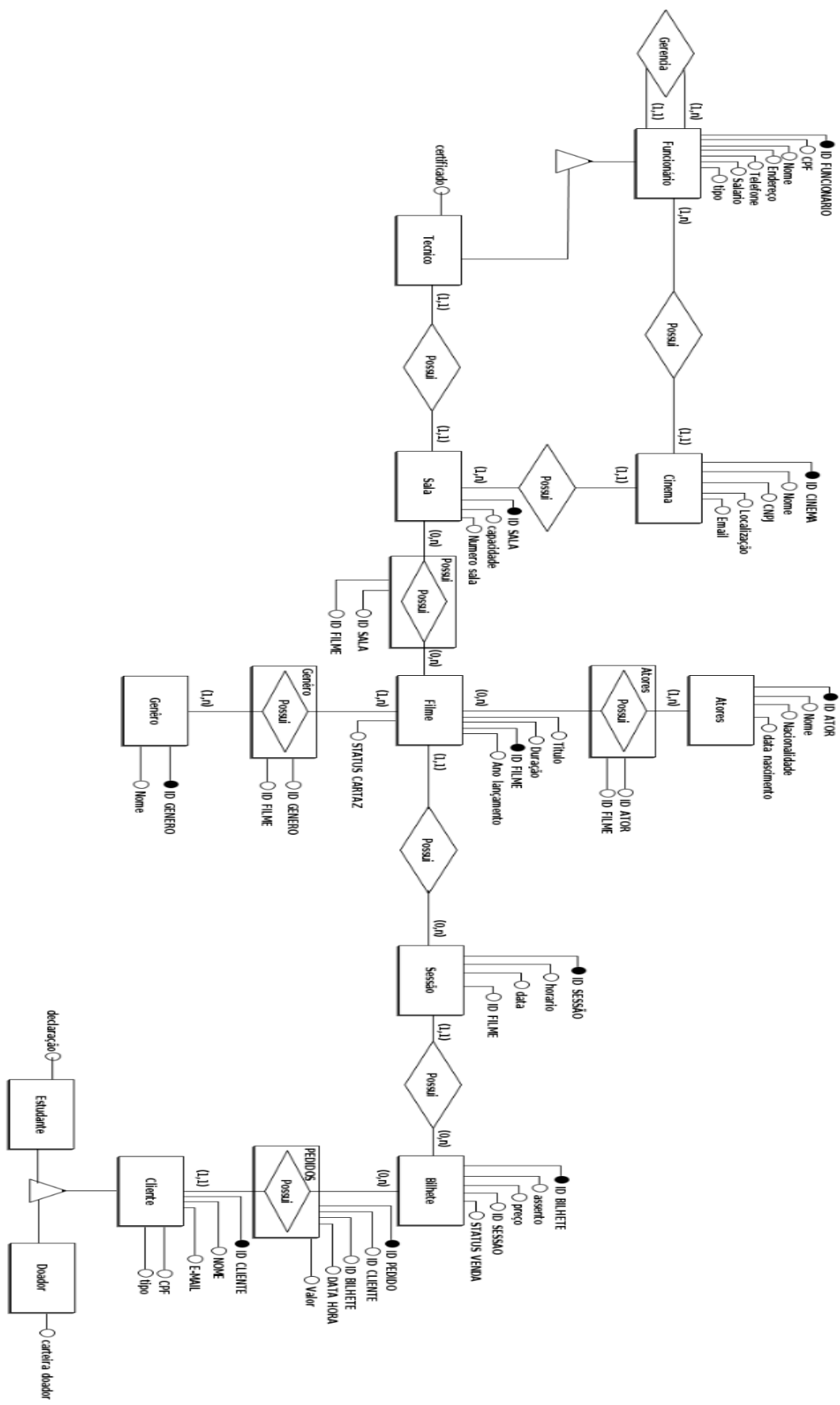


A tabela de salas, por sua vez, se relaciona com a tabela de filmes, indicando qual filme será exibido em cada sala. A tabela de filmes se relaciona com as tabelas de atores e de gênero, indicando quais atores participaram do filme e qual é o gênero do filme.

Além disso, a tabela de filmes se relaciona com a tabela de sessões, indicando em que horários cada filme será exibido. A tabela de sessões se relaciona com a tabela de bilhetes, indicando quantos bilhetes foram vendidos para cada sessão. E a tabela de bilhetes se relaciona com a tabela de pedidos, que é responsável por armazenar as informações do pedido do cliente e as informações do bilhete comprado por ele, esta tabela também será responsável pelo cálculo dos valores e possíveis descontos que possam ocorrer na compra do bilhete.

Por fim, a tabela de clientes possui duas especializações não exclusivas, podendo o cliente ser doador ou estudante, necessitando de carteirinha de doação e declaração de estudante, oferecendo desconto de 50% no valor do bilhete para esses casos. Essas especificações indicam que alguns clientes possuem direito a promoções, como é o caso dos estudantes e dos doadores. Essas promoções podem ser aplicadas em bilhetes de cinema, por exemplo.

2.2 DIAGRAMA ENTIDADE RELACIONAMENTO



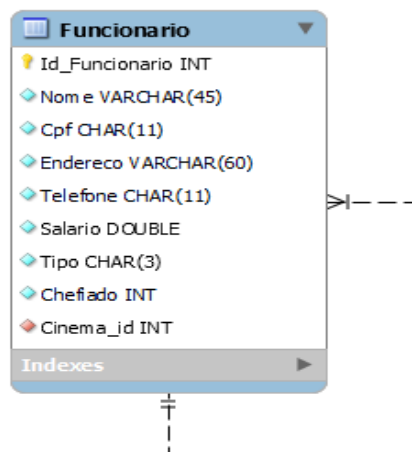
1 Figura 1: Modelo Entidade Relacionamento

### 3. MODELO LÓGICO

#### 3.1 MONTAGEM DO DIAGRAMA LÓGICO

##### ANÁLISE 1:

A abordagem adotada para o auto relacionamento do funcionário consiste em adicionar uma nova coluna denominada "CHEFIADO" à tabela existente, a qual armazena o identificador (ID) do funcionário responsável por supervisionar outros funcionários. Em vez de criar uma tabela separada para representar essa relação, a referência interna é estabelecida na mesma tabela, permitindo a associação direta entre o funcionário e o funcionário que ele chefia.



2

##### ENTIDADES E ATRIBUTOS:

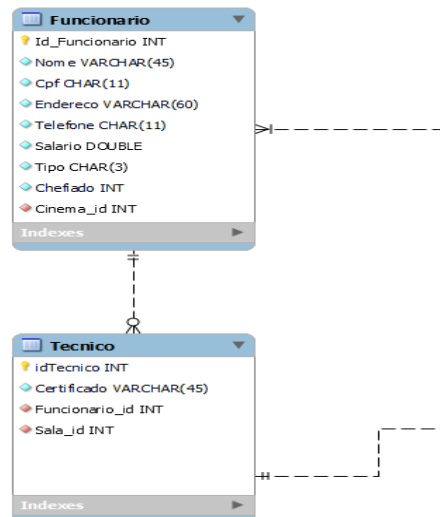
**FUNCIONÁRIO** (idFuncionario, Nome, Cpf, Endereço, Telefone, Salário, Tipo, Chefiado, Cinema\_id )

[Chefiado] referencia TÉCNICO

<sup>2</sup> Figura 2: Auto relacionamento de Funcionário

## ANÁLISE 2:

A tabela de funcionários apresenta uma especialização não exclusiva denominada "técnico", a qual o funcionário pode ser técnico de exibição de filmes, que se distingue dos demais cargos devido à presença do atributo "certificado", o qual impacta diretamente a remuneração dos funcionários nessa posição. A especialização é dita como 'não exclusiva', pois o funcionário pode apresentar outra função dentro do cinema.



3

## ENTIDADES E ATRIBUTOS:

**FUNCIONÁRIO** (idFuncionario, Nome, Cpf, Endereço, Telefone, Salário, Tipo, Chefiado, Cinema \_id )  
[Chefiado] referencia TÉCNICO

**TÉCNICO** (idTecnico, Certificado, Funcionário \_id, Sala \_id )  
[Funcionário \_id] referencia FUNCIONÁRIO

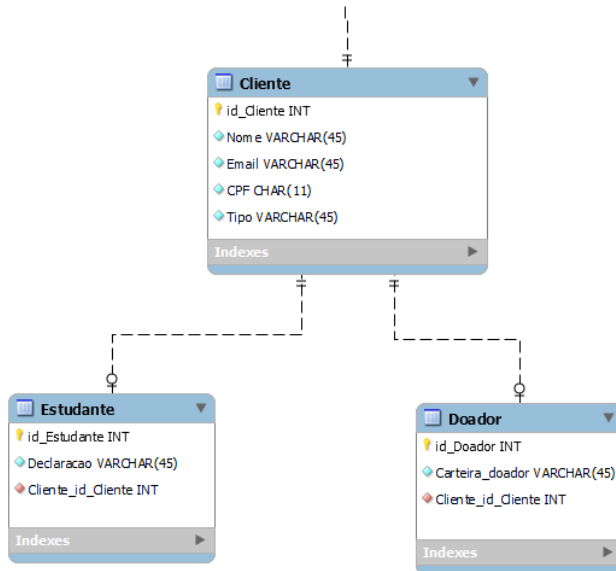
## ANÁLISE 3:

A tabela de clientes possui duas especializações não exclusivas: doador e estudante. Um cliente pode ser tanto doador quanto estudante, ou até mesmo ambas as coisas ao mesmo tempo. Para comprovar sua condição de doador, o cliente deve apresentar uma carteirinha de doação válida. Já para comprovar sua condição de estudante, é necessário apresentar uma declaração de estudante atualizada.

---

<sup>3</sup> Figura 3: Especificação de Funcionário

Como forma de benefício para essas categorias especiais, é oferecido um desconto de 50% no valor do bilhete para os clientes que se enquadram como doadores ou estudantes, ou seja, aqueles que comprovaram uma dessas condições. Esse desconto visa reconhecer e incentivar o engajamento do cliente como doador ou sua dedicação aos estudos, proporcionando uma vantagem financeira no acesso aos serviços ou produtos oferecidos.



4

## ENTIDADES E ATRIBUTOS

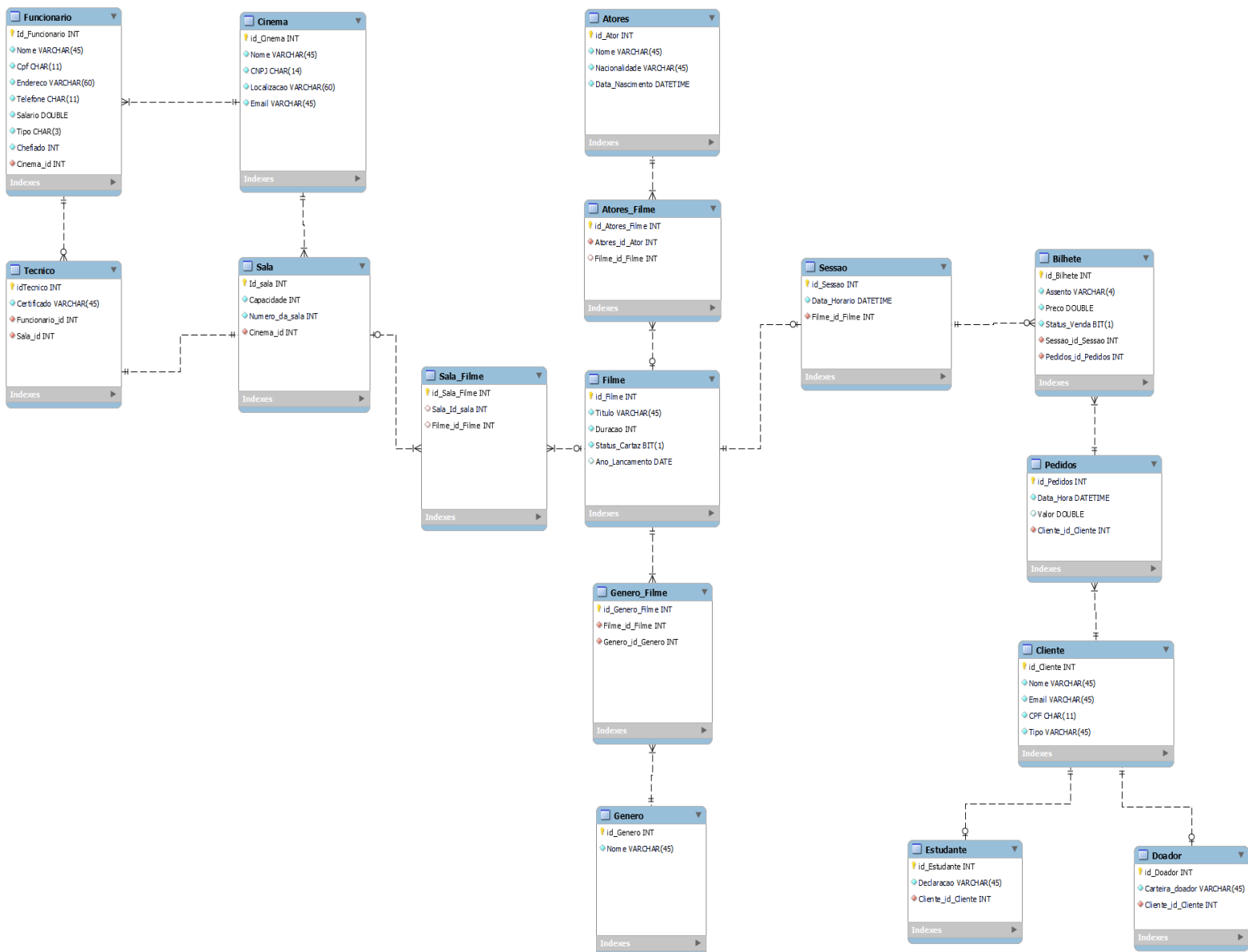
**CLIENTE** (id\_Cliente, Nome, Email, CPF, Tipo)

**ESTUDANTE** (id\_Estudante, Declaração, Cliente \_ id\_Cliente)  
[Cliente \_ id\_Cliente] referencia CLIENTE

**DOADOR** (id\_Doador, Carteira\_doador, Cliente \_ id\_Cliente)  
[Cliente \_ id\_Cliente] referencia CLIENTE

<sup>4</sup> Figura 4: Especificação de Clientes

## 3.2 DIAGRAMA LÓGICO



## 4. DICIONÁRIO DE DADOS

### 4.1 TABELA FUNCIONÁRIO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	PK	FK	UK
id_funcionario	int	não	codigo do funcionario		X		
nome	varchar(45)	não	nome do funcionário				
cpf	char(11)	não	cpf do funcionario				X
endereco	varchar(60)	não	endereço do funcionario				
telefone	char(11)	sim	telefone do funcionario				
salario	double	não	salário do funcionario				
tipo	char(3)	não	tipo do cargo				
chefiado	int	não	cod do funcionario que chefia				
cinema_id	int	não	codigo do cinema	tabela cinema		X	
TABELAS REFERENCIADAS POR FUNCIONARIO							
	> cinema -> fk: cinema_id. Relacionamento: 1,1						
TABELAS QUE REFERENCIAM FUNCIONÁRIO							
	> tecnico -> fk: funcionario_id. Relacionamento: 1,1						

Figura 6: Tabela de cadastro de funcionários do cinema

### 4.2 TABELA CINEMA

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	PK	FK	UK
id_cinema	int	não	codigo do cinema	-	X		
nome	varchar(45)	não	nome do cinema	-			
cnpj	char(14)	não	cnpj do cinema	-			X
localizacao	varchar(60)	não	localizacao do cinema	-			
email	varchar(45)	não	email do cinema	-			
TABELAS REFERENCIADAS POR CINEMA							
	> Nenhuma						
TABELAS QUE REFERENCIAM CINEMA							
	> Funcionário -> fk: cinema_id. Relacionamento: 1,N						
	> Sala -> fk: cinema id. Relacionamento: 1,N						

Figura 7: Tabela de cadastro das informações do cinema

### 4.3 TABELA FILME

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	PK	FK	UK
id_filme	int	não	codigo do filme	-	X		
titulo	varchar(45)	não	titulo do filme	-			X
duracao	int	não	duracao do filme	-			
status_cartaz	bit(1)	não	status do cartaz do filme	-			
ano_lancamento	date	não	ano de lancamento do filme	-			
TABELAS REFERENCIADAS POR FILME							
	> Nenhuma						
TABELAS QUE REFERENCIAM FILME							
	> Atores_filme -> fk: filme_id_filme. Relacionamento: 1,N						
	> Genero_filme -> fk: filme_id_filme. Relacionamento: 1,N						

Figura 8: Tabela de cadastro das informações dos filmes

### 4.4 TABELA CLIENTE

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	PK	FK	UK
id_cliente	int	não	codigo do cliente	-	X		
nome	varchar(45)	não	nome do cliente	-			
email	varchar(45)	sim	email do cliente	-			
cpf	char(11)	não	cpf do cliente	-			X
tipo	varchar(45)	não	tipo do cliente	-			
TABELAS REFERENCIADAS POR CLIENTE							
	> Nenhuma						
TABELAS QUE REFERENCIAM CLIENTE							
	> Pedidos -> fk: cliente_id_cliente. Relacionamento: 0,N						
	> Estudante -> fk: cliente_id_cliente. Relacionamento: 1,1						
	> Doador -> fk: cliente_id_cliente. Relacionamento: 1,1						

Figura 9: Tabela de cadastro das informações dos clientes do cinema

### 4.5 TABELA BILHETE

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	PK	FK	UK
id_bilhete	int	não	código do bilhete	-	X		
assento	varchar(4)	não	número do assento relacionado a capacidade da sala da sessão	-			
preco	double	não	preço do bilhete	-			
status_venda	bit(1)	não	status de true ou false para a venda do bilhete	-			
sessao_id	int	não	chave estrangeira que referencia a sessão do bilhete	tabela sessão		X	
pedidos_id	int	não	chave estrangeira que referencia o pedido do bilhete	tabela pedidos		X	

Figura 10: Tabela de cadastro do Bilhete



## 5. RESTRIÇÕES

### 5.1 TABELA FUNCIONÁRIO

<b>COLUNA</b>	<b>TIPO</b>	<b>NOME</b>	<b>EXPRESSÃO</b>
id_funcionario	Chave primária IDENTITY	PK_FUNC	PRIMARY KEY
cpf	Chave candidata	UK_CPF_FUNC	UNIQUE
cinema_id	Chave estrangeira referenciando a coluna cinema_id a tabela Cinema	FK_CINEMA	FOREIGN KEY

Figura 11: Tabela de restrições da tabela funcionário

### 5.2 TABELA CINEMA

<b>COLUNA</b>	<b>TIPO</b>	<b>NOME</b>	<b>EXPRESSÃO</b>
id_cinema	Chave primária IDENTITY	PK_CINEMA	PRIMARY
cnpj	Chave candidata	UK_CNPJ	UNIQUE

Figura 12: Tabela de restrições da tabela cinema

### 5.3 TABELA FILME

<b>COLUNA</b>	<b>TIPO</b>	<b>NOME</b>	<b>EXPRESSÃO</b>
Id_filme	Chave primária IDENTITY	PK_FILME	PRIMARY KEY
titulo	Chave candidata	UK_TITULO	UNIQUE

Figura 13: Tabela de restrições da tabela filme

### 5.4 TABELA CLIENTE

<b>COLUNA</b>	<b>TIPO</b>	<b>NOME</b>	<b>EXPRESSÃO</b>
id_cliente	Chave primária IDENTITY	PK_CLIENTE	PRIMARY KEY
cpf	Chave candidata	UK_CPF_CLIENTE	UNIQUE

Figura 14: Tabela de restrições da tabela cliente

## 5.5 TABELA BILHETE

<b>COLUNA</b>	<b>TIPO</b>	<b>NOME</b>	<b>EXPRESSÃO</b>
Id_bilhete	Chave primária IDENTITY	PK_FILME	PRIMARY KEY
Id_bilhete	Chave candidata	UK_BILHETE	UNIQUE
pedidos_id	Chave estrangeira referenciando a coluna pedidos_id a tabela Pedidos	FK_PEDIDOS	FOREIGN KEY
sessao_id	Chave estrangeira referenciando a coluna sessão_id a tabela Sessão	FK_SESSAO	FOREIGN KEY

Figura 15: Tabela de restrições da tabela bilhete

## 6. MODELO FÍSICO

### 6.1 CRIAÇÃO DO BANCO DE DADOS

```
create database Cinesercla;
```

### 6.2 SCRIPTS: CRIAÇÃO DE TABELAS, ATRIBUTOS E CONSTRAINTS

```
create table Cinema(  
    id_cinema int not null auto_increment,  
    nome varchar(45) not null,  
    CNPJ varchar(14) unique key not null,  
    localizacao varchar(60) not null,  
    email varchar (45) not null,  
    primary key(id_cinema)  
);
```

```
create table Sala(  
    id_sala int not null auto_increment,  
    capacidade int not null,  
    numero_da_sala int not null,  
    cinema_id int not null,  
    FOREIGN KEY(cinema_id) REFERENCES Cinema(id_cinema),  
    primary key(id_sala)
```

```
);
```

```
create table Funcionario(  
    id_funcionario int not null auto_increment,  
    nome varchar(45) not null,  
    cpf varchar(11) unique key not null,  
    endereco varchar(60) not null,  
    telefone char(11),  
    salario double not null,  
    tipo char(3) not null,  
    chefiado int not null,  
    cinema_id int not null,  
    FOREIGN KEY(cinema_id) REFERENCES Cinema(id_cinema),  
    primary key(id_funcionario)  
);
```

```
create table Tecnico(  
    id_tecnico int not null auto_increment,  
    certificado varchar(45) not null,  
    funcionario_id int not null,  
    FOREIGN KEY(funcionario_id) REFERENCES  
Funcionario(id_funcionario),  
    sala_id int not null,  
    FOREIGN KEY(sala_id) REFERENCES Sala(id_sala),  
    primary key(id_tecnico)  
);
```

```
create table Filme(  
    id_filme int not null auto_increment,  
    titulo varchar(45) unique key not null,  
    duracao time not null,  
    status_cartaz bit(1) not null,  
    ano_lancamento date not null,  
    primary key(id_filme)  
);
```

```
create table Sala_filme(  

```

```

    id_sala_filme int not null auto_increment,
    sala_id int not null,
    filme_id int not null,
    FOREIGN KEY(sala_id) REFERENCES Sala(id_sala),
    FOREIGN KEY(filme_id) REFERENCES Filme(id_filme),
    primary key(id_sala_filme)
);

create table Atores(
    id_ator int not null auto_increment,
    nome varchar(45) not null,
    nacionalidade varchar(45) not null,
    data_nascimento date not null,
    primary key(id_ator)
);

create table Atores_filme(
    id_ator_filme int not null auto_increment,
    ator_id int not null,
    filme_id int not null,
    FOREIGN KEY(ator_id) REFERENCES Atores(id_ator),
    FOREIGN KEY(filme_id) REFERENCES Filme(id_filme),
    primary key(id_ator_filme)
);

create table Genero(
    id_genero int not null auto_increment,
    nome varchar(45) not null,
    primary key(id_genero)
);

create table Genero_filme(
    id_genero_filme int not null auto_increment,
    genero_id int not null,
    filme_id int not null,
    FOREIGN KEY(genero_id) REFERENCES Genero(id_genero),
    FOREIGN KEY(filme_id) REFERENCES Filme(id_filme),

```

```

        primary key(id_genero_filme)
    );

create table Sessao(
    id_sessao int not null auto_increment,
    data_horario datetime not null,
    filme_id int not null,
    FOREIGN KEY(filme_id) REFERENCES Filme(id_filme),
    primary key(id_sessao)
);

create table Bilhete(
    id_bilhete int not null auto_increment,
    assento varchar(4) not null,
    preco double not null,
    status_venda bit(1) not null,
    sessao_id int not null,
    pedido_id int not null,
    FOREIGN KEY(sessao_id) REFERENCES Sessao(id_sessao),
    FOREIGN KEY(pedido_id) REFERENCES Pedidos(id_pedido),
    primary key(id_bilhete)
);

create table Pedidos(
    id_pedido int not null auto_increment,
    data_hora datetime not null,
    valor double not null,
    cliente_id int not null,
    FOREIGN KEY(cliente_id) REFERENCES Cliente(id_cliente),
    primary key(id_pedido)
);

create table Cliente(
    id_cliente int not null auto_increment,
    nome varchar(45) not null,
    email varchar(45) not null,
    cpf varchar(11) unique key not null,
    telefone char(11),

```

```

        tipo varchar(345) not null,
        primary key(id_cliente)
    );

create table Estudante(
    id_estudante int not null auto_increment,
    declaracao varchar(45) not null,
    cliente_id int not null,
    FOREIGN KEY(cliente_id) REFERENCES Cliente(id_cliente),
    primary key(id_estudante)
);

create table Doador(
    id_doador int not null auto_increment,
    carteira_doador varchar(45) not null,
    cliente_id int not null,
    FOREIGN KEY(cliente_id) REFERENCES Cliente(id_cliente),
    primary key(id_doador)
);

```

## 7. INSERÇÃO DE DADOS

### 7.1 INSERÇÃO DE VALORES NAS TABELAS

```

INSERT INTO Atores_filme (ator_id, filme_id) VALUES
    (1, 1),
    (3, 1),
    (2, 2),
    (4, 2),
    (5, 3),
    (6, 3),
    (7, 4),
    (8, 4),
    (9, 5),
    (10, 5),
    (11, 6),
    (12, 6),
    (13, 7),
    (14, 7),

```

```
(15, 7);
```

```
INSERT INTO Genero (nome) VALUES
```

```
('Ação'),  
('Comédia'),  
('Drama'),  
('Ficção Científica'),  
('Romance'),  
('Suspense'),  
('Terror');
```

```
INSERT INTO Genero_filme (genero_id, filme_id) VALUES
```

```
(1, 1),  
(4, 1),  
(3, 2),  
(7, 2),  
(3, 3),  
(6, 3),  
(4, 4),  
(6, 4),  
(2, 5),  
(3, 5),  
(4, 6),  
(6, 6),  
(5, 7),  
(6, 7),  
(7, 7);
```

```
INSERT INTO Sessao (data_horario, filme_id) VALUES
```

```
('2023-06-16 18:00:00', 1),  
('2023-06-16 20:30:00', 2),  
('2023-06-16 17:30:00', 3);
```

```
INSERT INTO Cliente (nome, email, cpf, telefone, tipo) VALUES
```

```
('João Silva', 'joao.silva@example.com', '12345678901',  
'999999999', 'Convencional'),  
('Maria Santos', 'maria.santos@example.com',  
'98765432109', '888888888', 'Doador');
```

```

        ('Pedro Oliveira', 'pedro.oliveira@example.com',
'45678912301', '777777777', 'Estudante');

INSERT INTO Estudante (declaracao, cliente_id) VALUES
('imgv2-2-f.scribdassets.com/img/document', 3);

INSERT INTO Doador (carteira_doador, cliente_id) VALUES
('imgv2-2-f.scribdassets.com/img/document', 2);

INSERT INTO Pedidos (data_hora, valor, cliente_id) VALUES
('2023-06-16 13:00:00', 40, 2),
('2023-06-14 11:00:00', 20, 1),
('2023-06-15 15:00:00', 60, 3);

INSERT INTO Bilhete (assento, preco, status_venda, sessao_id,
pedido_id) VALUES
('045', 20, 1, 1, 1),
('046', 20, 1, 1, 1),
('120', 20, 1, 2, 2),
('078', 20, 1, 3, 3),
('079', 20, 1, 3, 3),
('080', 20, 1, 3, 3);

```

## 7.2 APLICAÇÃO DE DESCONTO DA MEIA ENTRADA

```

UPDATE Pedidos
SET valor = valor / 2
WHERE cliente_id IN (SELECT id_cliente FROM Cliente WHERE tipo
= 'Estudante' OR tipo = 'Doador');

```

```

mysql>
mysql> UPDATE Pedidos
-> SET valor = valor / 2
-> WHERE cliente_id IN (SELECT id_cliente FROM Cliente WHERE tipo = 'Estudante' OR tipo = 'Doador');
Query OK, 2 rows affected (0,70 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from Pedidos;
+-----+-----+-----+-----+
| id_pedido | data_hora          | valor | cliente_id |
+-----+-----+-----+-----+
| 1 | 2023-06-16 13:00:00 | 20 | 2 |
| 2 | 2023-06-14 11:00:00 | 20 | 1 |
| 3 | 2023-06-15 15:00:00 | 30 | 3 |
+-----+-----+-----+-----+
3 rows in set (0,03 sec)

```

Figura 16 - Aplicação de desconto da meia entrada



## 8. CONSULTAS

### 8.1 CONSULTA SIMPLES

```
SELECT nome, localizacao FROM Cinema;
```

```
mysql> SELECT nome, localizacao FROM Cinema;
+-----+-----+
| nome      | localizacao |
+-----+-----+
| Cinesercla | Partage Shopping - Campina Grande |
+-----+-----+
1 row in set (2,76 sec)
```

Figura 17 - Consulta simples

### 8.2 CONSULTA COM JUNÇÃO

```
SELECT f.titulo, s.capacidade
FROM Filme f
JOIN Sala_filme sf ON f.id_filme = sf.filme_id
JOIN Sala s ON sf.sala_id = s.id_sala;
```

```
mysql>
mysql>
mysql> SELECT f.titulo, s.capacidade
-> FROM Filme f
-> JOIN Sala_filme sf ON f.id_filme = sf.filme_id
-> JOIN Sala s ON sf.sala_id = s.id_sala;
+-----+-----+
| titulo      | capacidade |
+-----+-----+
| Os Vingadores: Ultimato | 100 |
| Parasita    | 120 |
| Coringa     | 80 |
+-----+-----+
3 rows in set (0,35 sec)
```

Figura 18 - Consulta com junção

### 8.3 ALTERAÇÃO DE DADOS

```
UPDATE Cinema
SET nome = 'Novo Nome'
WHERE id_cinema = 1;
```

```
mysql> UPDATE Cinema
-> SET nome = 'Novo Nome'
-> WHERE id_cinema = 1;
Query OK, 1 row affected (0,63 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from Cinema;
+-----+-----+-----+-----+-----+
| id_cinema | nome      | CNPJ      | localizacao      | email      |
+-----+-----+-----+-----+-----+
| 1         | Novo Nome | 12345678901234 | Partage Shopping - Campina Grande | cinema@example.com |
+-----+-----+-----+-----+-----+
1 row in set (0,07 sec)
```

Figura 19 - Alteração de dados

## 8.4 EXCLUSÃO DE DADOS

```
DELETE FROM Funcionario
WHERE id_funcionario = 1;
```

```
mysql> select * from Funcionario;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_funcionario | nome      | cpf      | endereco      | telefone | salario | tipo | chefiado | cinema_id |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1             | João Silva | 12345678901 | Rua dos Funcionários, 123 | 111111111 | 2000 | Lim | 4 | 1 |
| 2             | Maria Santos | 98765432109 | Avenida dos Trabalhadores, 456 | 222222222 | 2500 | Atd | 4 | 1 |
| 3             | Pedro Oliveira | 56789012345 | Travessa das Ruas, 789 | 333333333 | 3000 | Tec | 4 | 1 |
| 4             | Ana Pereira | 01234567890 | Praça das Avenidas, 789 | 444444444 | 3500 | Ger | 4 | 1 |
| 5             | Carlos Rodrigues | 78901234567 | Rua das Flores, 789 | 555555555 | 3000 | Tec | 4 | 1 |
| 6             | Isabela Costa | 23456789012 | Avenida Central, 456 | 666666666 | 3000 | Tec | 4 | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0,06 sec)

mysql>
mysql> delete from Funcionario where id_funcionario = 1;
Query OK, 1 row affected (0,60 sec)

mysql> select * from Funcionario;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_funcionario | nome      | cpf      | endereco      | telefone | salario | tipo | chefiado | cinema_id |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2             | Maria Santos | 98765432109 | Avenida dos Trabalhadores, 456 | 222222222 | 2500 | Atd | 4 | 1 |
| 3             | Pedro Oliveira | 56789012345 | Travessa das Ruas, 789 | 333333333 | 3000 | Tec | 4 | 1 |
| 4             | Ana Pereira | 01234567890 | Praça das Avenidas, 789 | 444444444 | 3500 | Ger | 4 | 1 |
| 5             | Carlos Rodrigues | 78901234567 | Rua das Flores, 789 | 555555555 | 3000 | Tec | 4 | 1 |
| 6             | Isabela Costa | 23456789012 | Avenida Central, 456 | 666666666 | 3000 | Tec | 4 | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)
```

Figura 20 - Exclusão de dados

## 8.5 CONSULTA SIMPLES ENVOLVENDO UMA ÚNICA TABELA

```
SELECT * FROM Cinema;
```

```
mysql> select * from Cinema;
+-----+-----+-----+-----+-----+
| id_cinema | nome      | CNPJ      | localizacao      | email      |
+-----+-----+-----+-----+-----+
| 1         | Novo Nome | 12345678901234 | Partage Shopping - Campina Grande | cinema@example.com |
+-----+-----+-----+-----+-----+
1 row in set (1,04 sec)
```

Figura 21 - Consulta simples envolvendo uma única tabela

## 8.6 CONSULTA SIMPLES ENVOLVENDO DUAS TABELAS

```
SELECT f.titulo, s.capacidade
FROM Filme f
JOIN Sala_filme sf ON f.id_filme = sf.filme_id
JOIN Sala s ON sf.sala_id = s.id_sala;
```

```
mysql> SELECT f.titulo, s.capacidade
-> FROM Filme f
-> JOIN Sala_filme sf ON f.id_filme = sf.filme_id
-> JOIN Sala s ON sf.sala_id = s.id_sala;
+-----+-----+
| titulo          | capacidade |
+-----+-----+
| Os Vingadores: Ultimato |      100 |
| Parasita        |      120 |
| Coringa         |       80 |
+-----+-----+
3 rows in set (0,14 sec)
```

Figura 22 - Consulta simples envolvendo duas tabelas

## 8.7 CONSULTA SIMPLES ENVOLVENDO TRÊS TABELAS

```
SELECT c.nome, f.titulo, s.numero_da_sala
FROM Cinema c
JOIN Sala s ON c.id_cinema = s.cinema_id
JOIN Sala_filme sf ON s.id_sala = sf.sala_id
JOIN Filme f ON sf.filme_id = f.id_filme;
```

```
mysql> SELECT c.nome, f.titulo, s.numero_da_sala
-> FROM Cinema c
-> JOIN Sala s ON c.id_cinema = s.cinema_id
-> JOIN Sala_filme sf ON s.id_sala = sf.sala_id
-> JOIN Filme f ON sf.filme_id = f.id_filme;
+-----+-----+-----+
| nome      | titulo          | numero_da_sala |
+-----+-----+-----+
| Novo Nome | Os Vingadores: Ultimato |      1 |
| Novo Nome | Parasita        |      2 |
| Novo Nome | Coringa         |      3 |
+-----+-----+-----+
3 rows in set (0,05 sec)
```

Figura 23 - Consulta simples envolvendo três tabelas

## 8.8 CONSULTA ENVOLVENDO AGRUPAMENTO

```

SELECT f.titulo, COUNT(sf.id_sala_filme) AS quantidade_salas
FROM Filme f
JOIN Sala_filme sf ON f.id_filme = sf.filme_id
GROUP BY f.titulo;

```

```

mysql> SELECT f.titulo, COUNT(sf.id_sala_filme) AS quantidade_salas
-> FROM Filme f
-> JOIN Sala_filme sf ON f.id_filme = sf.filme_id
-> GROUP BY f.titulo;
+-----+-----+
| titulo                | quantidade_salas |
+-----+-----+
| Os Vingadores: Ultimato | 1                |
| Parasita                | 1                |
| Coringa                 | 1                |
+-----+-----+
3 rows in set (0,08 sec)

```

Figura 24 - Consulta envolvendo agrupamento

## 8.9 CONSULTA ENVOLVENDO FUNÇÃO AGREGADA

```

SELECT f.titulo, AVG(b.preco) AS media_preco
FROM Filme f
JOIN Sessao s ON f.id_filme = s.filme_id
JOIN Bilhete b ON s.id_sessao = b.sessao_id
GROUP BY f.titulo;

```

```

mysql> SELECT f.titulo, AVG(b.preco) AS media_preco
-> FROM Filme f
-> JOIN Sessao s ON f.id_filme = s.filme_id
-> JOIN Bilhete b ON s.id_sessao = b.sessao_id
-> GROUP BY f.titulo;
+-----+-----+
| titulo                | media_preco      |
+-----+-----+
| Os Vingadores: Ultimato | 20               |
| Parasita                | 20               |
| Coringa                 | 20               |
+-----+-----+
3 rows in set (0,13 sec)

```

Figura 25 - Consulta envolvendo função agregada



