

Faculdade De Engenharia da Universidade do Porto



Licenciatura em Engenharia Informática e Computação

Gestão de uma cadeia de cinemas

Bases de Dados

Docentes:

Carla Alexandra Teixeira Lopes

Michel Celestino Paiva Ferreira

Luis Pedro da Silva Couto

2LEIC11

Grupo 1105

Estudantes e Autores:

Tomás Malveiro do Carmo up202007590@fc.up.pt

José Miguel Araújo Ribeiro up202007231@fe.up.pt

Bruna Brasil Leão Marques up202007191@fe.up.pt

Índice

Modelo Conceptual	3
1.1 Contexto	3
1.2 Diagrama UML	4
Esquema Relacional	6
Análise de Dependências Funcionais e Formas Normais	7
3.1 Dependências Funcionais	7
3.2 Formas Normais	8
Restrições	9

1. Modelo Conceptual

1.1 Contexto

Este trabalho consiste em armazenar e organizar a informação relativa a uma rede de cinemas.

Em primeiro lugar, cada cinema possui um id para identificação, nome, morada, contacto e é composto por 5 a 10 salas, sendo que poderão ser de diferentes tipos, por exemplo, *IMAX*. Por sua vez, a sala também é identificada por um id e é constituída por um número específico de lugares. As salas nesta cadeia de cinemas possuem no mínimo 50 lugares e no máximo 200.

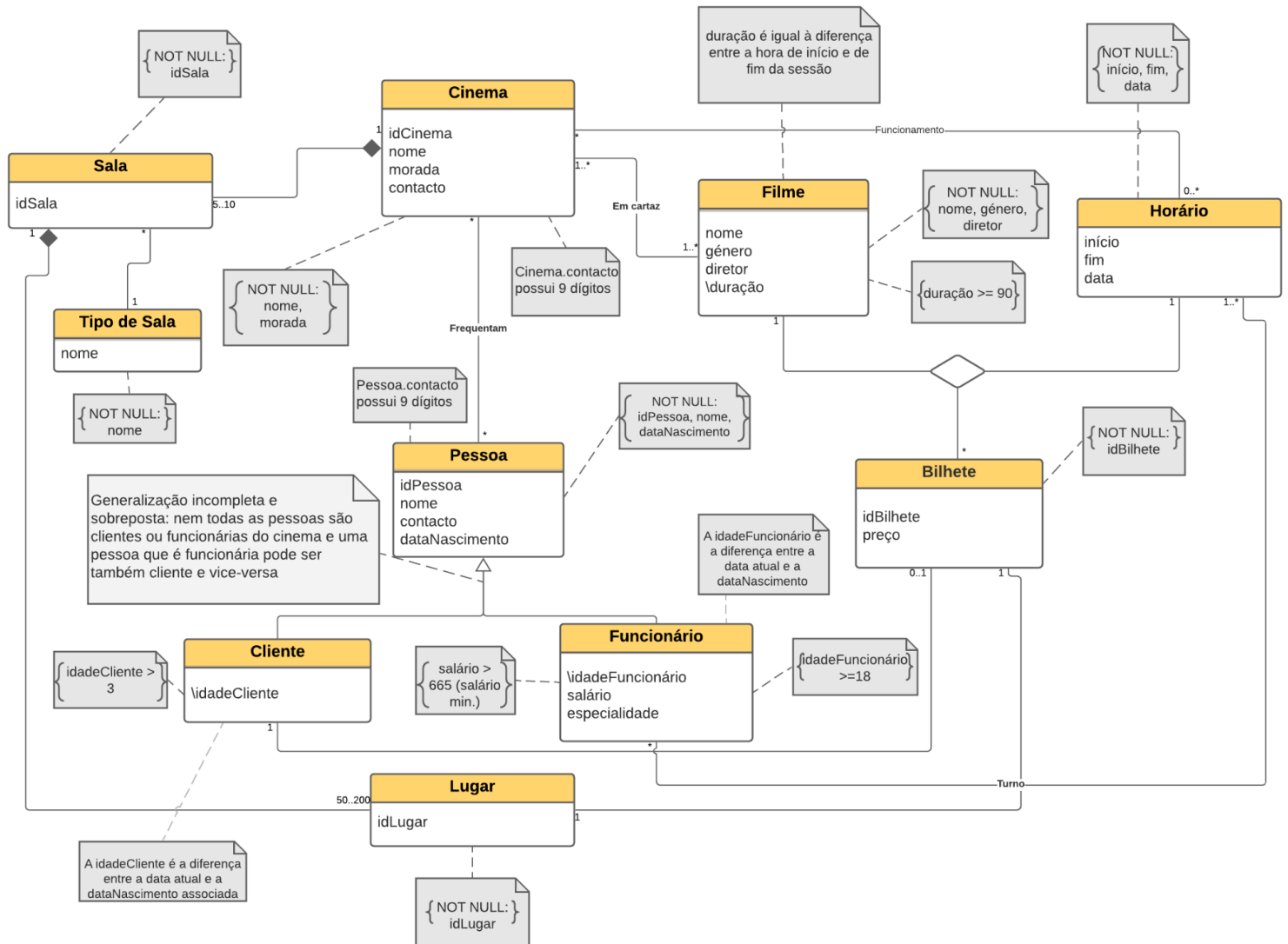
Sobre as pessoas que frequentam o cinema é registado o nome, contacto, um id e a sua data de nascimento. Uma pessoa sempre será um cliente ou um funcionário. A idade é diferenciada para os funcionários e clientes, uma vez que os funcionários são todos maiores de idade (18 anos ou mais) e o cinema só permite a entrada de clientes que têm mais de 3 anos.

O cinema emprega funcionários que trabalham numa certa especialidade e recebem um salário (maior ou igual ao salário mínimo). Relativamente aos clientes do cinema, cada um comprará um bilhete que indicará o preço, o lugar numa sala, o filme a ser visto e o horário da sessão.

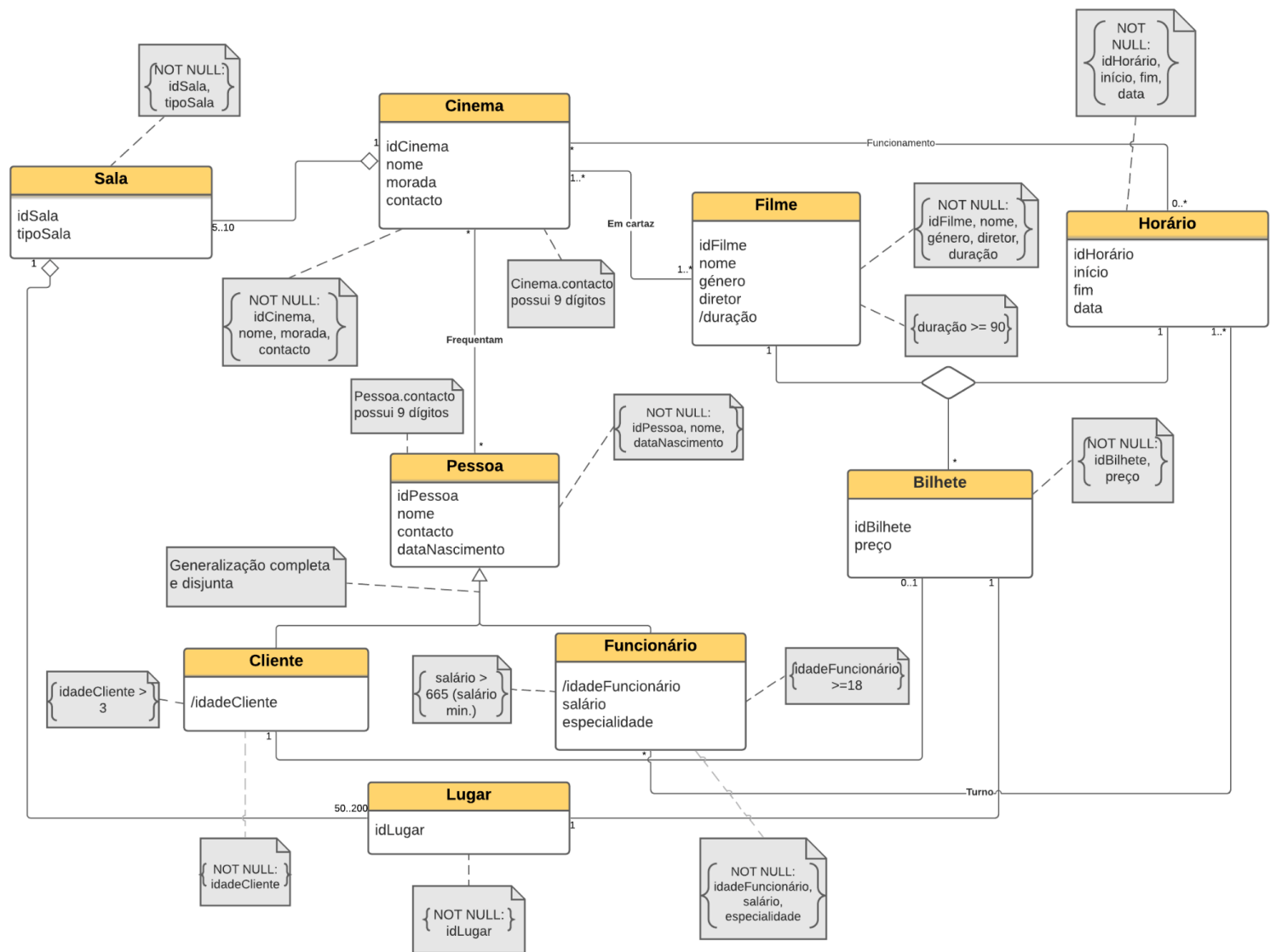
Para cada filme em cartaz no cinema guarda-se o seu id, nome, duração, diretor e género. Os filmes em exibição têm no mínimo uma duração de 90 minutos.

Todos os horários, quer de uma sessão de um filme quer do turno de trabalho de um funcionário quer do cinema possuem um id, uma hora de início, fim e a data do dia, mês e ano respetivo.

1.2 Diagrama UML



1.3 Diagrama UML Revisto



2. Esquema Relacional

Cinema (idCinema, nome, morada, contacto)

Cliente (idPessoa, nome, contacto, dataNascimento, idadeCliente)

Funcionário (idPessoa, nome, contacto, dataNascimento, idadeFuncionário, salário, especialidade)

CinemaCliente (idCinema -> Cinema, idPessoa -> Cliente)

CinemaFuncionário (idCinema -> Cinema, idPessoa -> Funcionário)

Horário (idHorário, início, fim, data)

Filme (idFilme, nome, género, diretor, duração)

EmCartaz (idFilme -> Filme, idCinema -> Cinema)

Funcionamento (idCinema -> Cinema, idHorário -> Horário)

Turno (idPessoa -> Funcionario, idHorário -> Horário)

Bilhete (idBilhete, preço, idPessoa -> Cliente, idLugar -> Lugar)

FilmeHorárioBilhete (idFilme -> Filme, idHorário -> Horário, idBilhete -> Bilhete)

Lugar (idLugar, idSala -> Sala)

Sala (idSala, tipoSala, idCinema -> Cinema)

Nota: Para representar a generalização disjunta e completa optou-se por estratégia de conversão: “object-oriented” sendo que esta é recomendada para generalizações disjuntas.

3. Análise de Dependências Funcionais e Formas Normais

3.1 Dependências Funcionais

- Cinema (idCinema, nome, morada, contacto)
{idCinema} -> {nome, morada, contacto}
- Cliente (idPessoa, nome, contacto, dataNascimento, idadeCliente)
{idPessoa} -> {nome, contacto, dataNascimento, idadeCliente}
- Funcionário (idPessoa, nome, contacto, dataNascimento, idadeFuncionário, salário, especialidade)
{idPessoa} -> {nome, contacto, dataNascimento, idadeFuncionário, salário, especialidade}
- CinemaCliente (idCinema -> Cinema, idPessoa -> Cliente)
{idCinema, idPessoa} -> {idCinema, idPessoa}
- CinemaFuncionário (idCinema -> Cinema, idPessoa -> Funcionário)
{idCinema, idPessoa} -> {idCinema, idPessoa}
- Horário (idHorário, início, fim, data)
{idHorário} -> {início, fim, data}
{início, fim, data} -> {idHorário}
- Filme (idFilme, nome, género, diretor, duração)
{idFilme} -> {nome, género, diretor, duração}
{nome, diretor} -> {idFilme}
- EmCartaz (idFilme -> Filme, idCinema -> Cinema)
{idFilme, idCinema} -> {idFilme, idCinema}
- Funcionamento (idCinema -> Cinema, idHorário -> Horário)
{idCinema, idHorário} -> {idCinema, idHorário}
- Turno (idPessoa -> Funcionário, idHorário -> Horário)
{idPessoa, idHorário} -> {idPessoa, idHorário}

- Bilhete (idBilhete, preço, idPessoa -> Cliente, idLugar -> Lugar)
{idBilhete} -> {preço, idPessoa, idLugar}
- FilmeHorárioBilhete (idFilme -> Filme, idHorário -> Horário, idBilhete -> Bilhete)
{idFilme, idHorário, idBilhete} -> {idFilme, idHorário, idBilhete}
- Lugar (idLugar, idSala -> Sala)
{idLugar} -> {idSala}
- Sala (idSala, tipoSala, idCinema -> Cinema)
{idSala} -> {tipoSala, idCinema}

3.2 Formas Normais

Uma relação está em BCNF se, para cada DF (dependência funcional) $A \rightarrow B$, ou esta dependência é trivial ou A é uma (super)chave.

Observa-se que nenhuma relação viola a BCNF.

Se para cada DF $A \rightarrow B$ não trivial, A é uma (super)chave ou B consiste apenas em atributos primos (atributos que estão presentes em qualquer uma das chaves candidatas) ou caso não tenha dependências funcionais não triviais, temos uma relação na 3FN (terceira forma normal).

Deste modo, observamos que todas as relações estão na 3FN.

4. Restrições

Cinema

- idCinema PRIMARY KEY - Não pode haver dois cinemas com o mesmo id e a partir deste temos acesso ao seu nome, morada e contacto;
- contacto UNIQUE - Não há números de contacto repetidos entre os cinemas;
- morada UNIQUE - Não há cinemas com a mesma morada;
- contacto CHECK (LENGTH(contacto) == 9) - Os contactos têm 9 dígitos;
- Na classe Cinema, os atributos nome, morada e contacto são definidos como NOT NULL uma vez que todos os cinemas necessitam desses atributos para existir:
 - nome NOT NULL, morada NOT NULL, contacto NOT NULL.

Cliente

- idPessoa PRIMARY KEY - Não pode haver pessoas diferentes com o mesmo id e a partir deste temos acesso ao seu nome, contacto e data de nascimento;
- contacto UNIQUE - Não há pessoas diferentes com o mesmo número de contacto;
- nome, dataNascimento e contacto são definidos como NOT NULL, pois é necessário estas informações para os clientes serem registados:
 - nome NOT NULL, dataNascimento NOT NULL, contacto NOT NULL
- contacto CHECK (LENGTH(contacto) == 9) - Os contactos têm 9 dígitos;
- idadeCliente CHECK (idadeCliente > 3) - O cliente do cinema deve ter idade maior que 3 anos.

Funcionário

- idPessoa PRIMARY KEY - Não pode haver pessoas diferentes com o mesmo id e a partir deste temos acesso ao seu nome, contacto e data de nascimento;
- contacto UNIQUE - Não há pessoas diferentes com o mesmo número de contacto;
- nome, dataNascimento, contacto e especialidade são definidos como NOT NULL, pois todos os funcionários necessitam destes atributos para serem registadas:
 - nome NOT NULL, dataNascimento NOT NULL, contacto NOT NULL
- contacto CHECK (LENGTH(contacto) == 9) - Os contactos têm 9 dígitos;
- idadeFuncionario CHECK (idadeFuncionario >= 18) - O funcionário do cinema deve ter idade maior ou igual que 18 anos;
- salario CHECK (salario > 665) - O salário do funcionário do cinema deve ser maior que o salário mínimo (considerado neste caso como 665).

CinemaCliente

- PRIMARY KEY (idCinema, idPessoa) - Não pode haver repetição do par (idCinema, idPessoa);
- idCinema REFERENCES Cinema(idCinema) - O id do cinema corresponde ao id de uma instância da tabela Cinema;
- idPessoa REFERENCES Cliente(idPessoa) - O id de pessoa corresponde ao id de uma instância da tabela Cliente.

CinemaFuncionario

- PRIMARY KEY (idCinema, idPessoa) - Não pode haver dois pares destes iguais e trivialmente a partir idCinema e idPessoa, temos estes mesmos atributos;
- idCinema REFERENCES Cinema(idCinema) - O id do cinema corresponde ao id de uma instância da tabela Cinema;
- idPessoa REFERENCES Funcionario(idPessoa) - O id de pessoa corresponde ao id de uma instância da tabela Funcionario.

Horário

- idHorario PRIMARY KEY - Não pode haver horários diferentes com o mesmo id e a partir deste temos acesso à hora de início, de fim e a data;
- Os atributos data, inicio e fim são definidos como NOT NULL, pois a todos os horários devem estar associados uma data, hora de início e hora de fim:
 - data NOT NULL, inicio NOT NULL, fim NOT NULL

Filme

- idFilme PRIMARY KEY - Não pode haver dois filmes com o mesmo id;
- duracao CHECK (duracao>90) - A duração de cada filme é de no mínimo 90 minutos;
- nome, genero e diretor são definidos como NOT NULL, pois é necessário estas informações para um filme ser registrado.
 - nome NOT NULL, genero NOT NULL, diretor NOT NULL, duracao NOT NULL.

EmCartaz

- PRIMARY KEY(idFilme, idCinema) - Não pode haver repetição do par (idFilme, idCinema);
- idFilme REFERENCES Filme(idFilme) - O idFilme corresponde ao id de uma instância da tabela Filme;
- idCinema REFERENCES Cinema(idCinema) - O id do cinema corresponde ao id de uma instância da tabela Cinema.

Funcionamento

- PRIMARY KEY(idCinema, idHorario) - Não pode haver repetição do par (idCinema, idHorario);

- idCinema REFERENCES Cinema(idCinema) - O id do cinema corresponde ao id de uma instância da tabela Cinema;
- idHorario REFERENCES Horario(idHorario) - O id do horário corresponde a um id da tabela Horário.

Turno

- PRIMARY KEY(idPessoa, idHorario) - Não pode haver repetição do par (idPessoa, idHorario);
- idPessoa REFERENCES Funcionario(idPessoa) - O id de pessoa corresponde ao id de uma instância da tabela Funcionario;
- idHorario REFERENCES Horario(idHorario) - O id do horário corresponde a um id da tabela Horário.

Bilhete

- idBilhete PRIMARY KEY - Não há dois bilhetes com o mesmo id;
- idPessoa REFERENCES Cliente(idPessoa) - O id de pessoa corresponde ao id de uma instância da tabela Cliente;
- idLugar REFERENCES Lugar(idLugar) - O id de um lugar corresponde a um id na tabela Lugar.

FilmeHorarioBilhete

- idFilme REFERENCES Filme(idFilme) - O id de filme corresponde ao id de uma instância da tabela Filme;
- idHorario REFERENCES Horario(idHorario) - O id do horário corresponde a um id da tabela Horário;
- idBilhete REFERENCES Bilhete(idBilhete) - O id do bilhete corresponde a um id da tabela Bilhete;
- PRIMARY KEY(idFilme, idHorario, idBilhete) - Não pode haver repetição do trio (idFilme, idHorario, idBilhete).

Lugar

- idLugar PRIMARY KEY - Não pode haver dois lugares com o mesmo id;
- idSala REFERENCES Sala(idSala) - O id da sala de cinema corresponde a um id da tabela Sala;

Sala

- idSala PRIMARY KEY - Não pode haver duas salas com o mesmo id;
- idCinema REFERENCES Cinema - O id do cinema corresponde a um id da tabela cinema;
- tipoSala NOT NULL - Todas as salas tem o seu tipo.

Contribuições:

Tomás Carmo: 33%

Bruna: 33%

José Miguel: 33%