

Grupo 103

Cadeia de Cinemas

Bases de Dados - 2019/2020

Beatriz Costa Silva Mendes up201806551@fe.up.pt

Daniel Gomes Silva up201909935@fe.up.pt

Teresa Isabel da Silva Corado up201806479@fe.up.pt

1 Contexto

1.1 Cinema

Um cinema é composto por várias bilheteiras e várias salas, onde são exibidos os filmes em cartaz. Relativamente a um cinema é necessário conhecer o seu nome, localidade e horas de abertura e fecho.

Num cinema estão presentes tanto clientes como funcionários. Dentro dos funcionários, existe um e um só gerente.

Cada cinema possui várias salas de visualização identificadas por um número. Estas têm uma capacidade e um tipo, isto é, se transmitem filmes em IMAX, 3D, etc., e estão associadas a um e um só filme, possuindo várias sessões para este ao longo do dia. Deste modo, é necessário saber as vagas que existem em cada sala em determinadas sessões.

1.2 Pessoa

Cada pessoa (cliente ou funcionário) é identificada pelo número do Cartão de Cidadão, pelo seu nome e idade. Os funcionários, ao contrário dos clientes, ainda são caracterizados por um ID, um salário e pelas suas horas de trabalho semanais. Cada funcionário pode trabalhar num único cinema e é, obrigatoriamente, maior de idade.

Os clientes distinguem-se através do seu estatuto (normal, estudante, sénior ou juvenil). Alguns destes podem usufruir de descontos nos bilhetes, o que influenciará o seu preço. Um cliente poderá comprar um ou mais bilhetes e snacks na bilheteira, onde o valor total de vendas é guardado.

1.3 Bilheteira (Bilhetes e Snacks)

Um bilhete é caracterizado pelo preço, lugar (na sala) e a sessão a que dá acesso. O preço deste varia tendo em conta o estatuto do cliente que vai usufruir do bilhete e no tipo da sala onde vai ser exibido o filme.

Um snack (tanto comida como bebida) é identificado por um nome e um preço. O cliente poderá optar por comprar snacks individualmente ou por comprar um menu. O menu é constituído apenas por bebida e pipocas e o seu preço será inferior ao preço de venda dos snacks individualmente.

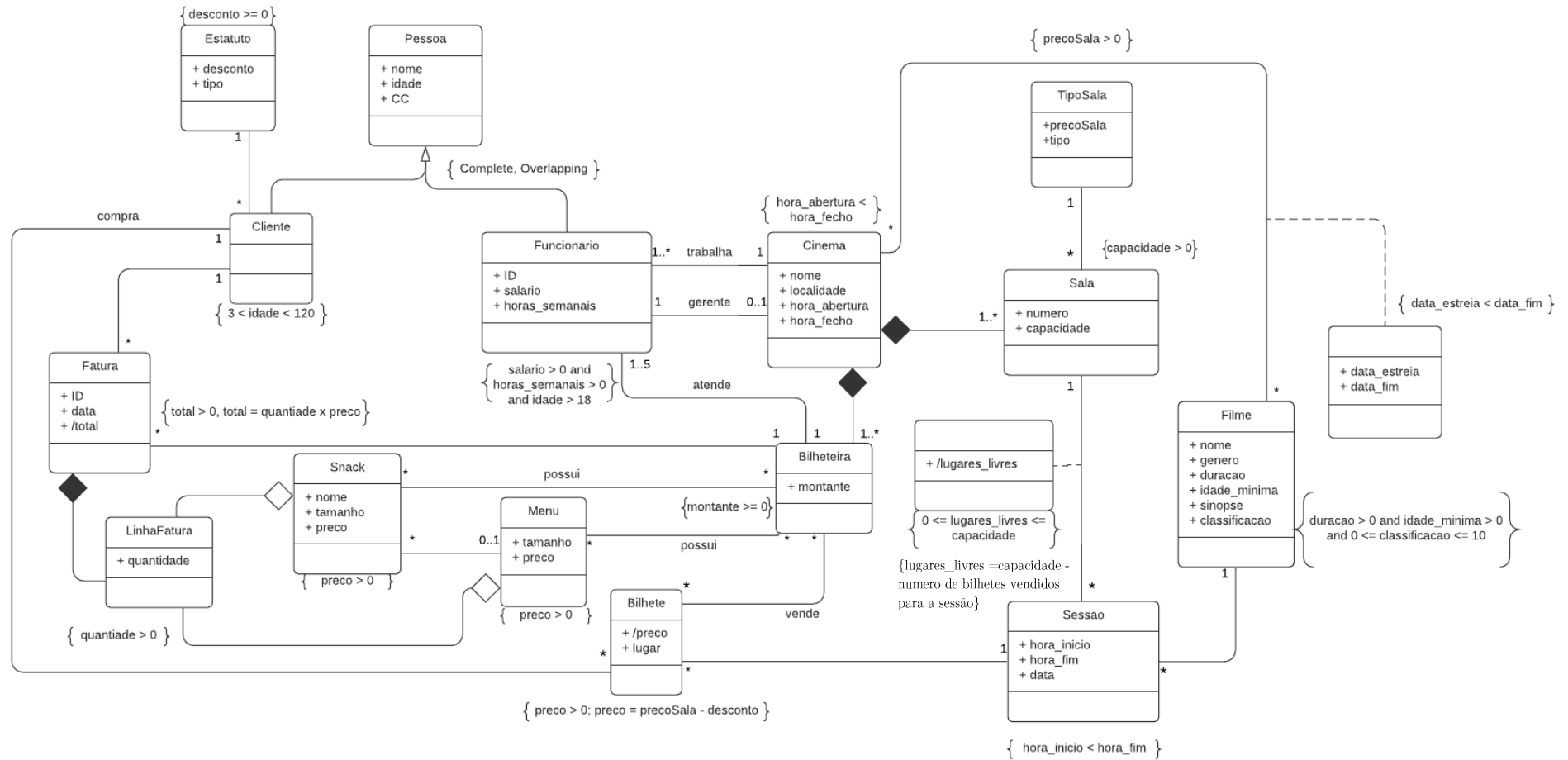
Deve existir uma fatura caracterizada por um ID, uma data e um total. Este consiste na soma dos preços dos snacks e dos menus comprados multiplicados pela quantidade respetiva.

Numa bilheteira existe sempre pelo menos 1 e no máximo 5 funcionários.

1.4 Filme

Um filme está associado a uma sala e possui várias sessões diárias, caracterizadas pelas horas de início e de fim. De cada um pretende-se armazenar o seu nome, género, duração, idade mínima para assistir, classificação e sinopse. A sua exibição realiza-se entre a data de estreia e data de fim de exibição.

2 Diagrama UML



3 Modelo Relacional e Dependências Funcionais

Nota: Na conversão da generalização utilizou-se a estratégia E/R style, uma vez que, esta é uma das melhores para generalizações Overlapping.

Pessoa(idPessoa, nome, idade, CC)

1. $\text{idPessoa} \rightarrow \text{nome}, \text{idade}, \text{CC}$

Cliente(idPessoa \rightarrow Pessoa, idEstatuto \rightarrow Estatuto)

1. $\text{idPessoa} \rightarrow \text{idEstatuto}$

Estatuto(idEstatuto, desconto, tipo)

1. $\text{idEstatuto} \rightarrow \text{desconto}, \text{tipo}$

Funcionario(idPessoa \rightarrow Pessoa, ID, salario, horas_semanais, idCinema \rightarrow Cinema, idBilheteira \rightarrow Bilheteira)

1. $\text{idPessoa} \rightarrow \text{ID}, \text{salario}, \text{horas_semanais}, \text{idCinema}, \text{idBilheteira}$

Cinema(idCinema, nome, localidade, hora_abertura, hora_fecho, idPessoa \rightarrow Funcionario)

1. $\text{idCinema} \rightarrow \text{nome}, \text{localidade}, \text{hora_abertura}, \text{hora_fecho}, \text{idPessoa}$

Bilheteira(idBilheteira, montante, idCinema \rightarrow Cinema)

1. $\text{idBilheteira} \rightarrow \text{montante}, \text{idCinema}$

Bilhete(idBilhete, preco, lugar, idPessoa \rightarrow Cliente, idSessao \rightarrow Sessao)

1. $\text{idBilhete} \rightarrow \text{preco}, \text{lugar}, \text{idPessoa}, \text{idSessao}$

Snack(idSnack, nome, tamanho, preco, idMenu \rightarrow Menu)

1. $\text{idSnack} \rightarrow \text{nome}, \text{tamanho}, \text{preco}, \text{idMenu}$

Menu(idMenu, tamanho, preco)

1. $\text{idMenu} \rightarrow \text{tamanho}, \text{preco}$

Sessao(idSessao, hora_inicio, hora_fim, data, idFilme→Filme, idSala→Sala, lugares_livres)

1. idSessao → hora_inicio, hora_fim, data, idFilme, idSala, lugares_livres

Sala(idSala, numero, capacidade, idTipoSala→TipoSala, idCinema→Cinema)

1. idSala → numero, capacidade, idTipoSala, idCinema

TipoSala(idTipoSala, precoSala, tipo)

1. idTipoSala → precoSala, tipo

Filme(idFilme, nome, genero, duracao, idade_minima, sinopse, classificacao)

1. idFilme → nome, genero, duracao, idade_minima, sinopse, classificacao, idSessao, idCinema

CinemaFilme(idFilme→Filme, idCinema→Cinema, data_estreia, data_fim)

1. idCinema, idFilme → data_estreia, data_fim

Fatura(idFatura, data, total, idBilheteira→Bilheteira, idPessoa→Cliente)

1. idFatura → data, total, idBilheteira, idPessoa

LinhaFatura(idLinhaFatura, quantidade, idSnack→Snack, idMenu→Menu, idFatura→Fatura)

1. idLinhaFatura → quantidade, idSnack, idMenu, idFatura

BilheteiraMenu(idBilheteira→Bilheteira, idMenu→Menu)

BilheteiraSnack(idBilheteira→Bilheteira, idSnack→Snack)

BilheteiraBilhete(idBilheteira→Bilheteira, idBilhete→Bilhete)

4 Formas Normais

De forma a prevenir o possível aparecimento de anomalias nos dados, é necessário assegurar o cumprimento das Formas Normais lecionadas: Forma Normal Boyce-Codd (BCNF) e 3ª Forma Normal (3NF). Assim sendo, foi necessário durante o desenvolvimento do projeto ter em atenção algumas premissas que estabelecem o cumprimento ou não destas Formas Normais.

Forma Normal Boyce-Codd:

Uma relação R com dependências funcionais FD está na Forma Normal Boyce-Codd (BCNF) se:

- Para cada dependência funcional não trivial $\overline{A} \rightarrow \overline{B}$, \overline{A} é uma (super)chave.

3ª Forma Normal

Uma relação R com dependências funcionais FD está na 3ª Forma Normal se:

- Para cada dependência funcional não trivial $\overline{A} \rightarrow \overline{B}$ se se verificar pelo menos uma das seguintes propriedades:
 - \overline{A} é uma (super)chave (BCNF)
 - \overline{B} consiste apenas de atributos primos (atributos que pertencem a uma chave)

Assim sendo, pode-se afirmar que se uma relação estiver na Forma Normal Boyce-Codd, encontra-se também na 3ª Forma Normal, uma vez que as 3NF são um *superset* das BCNF.

No projeto desenvolvido pelo nosso grupo de trabalho, todas as relações encontram-se nas Formas Normais, mais concretamente na BCNF, uma vez que na implementação de todas elas é adicionado um identificador (por exemplo, em Filme existe um idFilme) que funciona como chave da relação. Assim sendo, todas as relações acabam por gerar no máximo uma dependência funcional, garantindo-se assim a primeira premissa mencionada, que tanto prova que as relações estão na BCNF e na 3NF, pois, tal como afirmado anteriormente, as 3NF são um *superset* das BCNF.

5 Restrições

Cinema(idCinema, nome, localidade, hora_abertura, hora_fecho, idPessoa→Funcionario)

- Cada cinema tem um id único, idCinema (**primary key**).
- Cada cinema deve ter um número de identificação, nome, localidade, horas de abertura e de fecho e gerente (todos **not null**).
- idPessoa representa o gerente do cinema, funcionando como **foreign key** para um funcionário.
- hora_abertura deve ser superior a 10:00.
- hora_fecho deve encontrar-se entre as 00:00 e as 03:00.

Pessoa(idPessoa, nome, idade, CC)

- Cada pessoa deve ter um número de identificação, nome, idade e número de cartão de cidadão (todos **not null**).
- Cada pessoa tem um ID único (**primary key**).
- Os números de cartão de cidadão devem ser únicos (**unique**).
- idade deve estar entre 3 e 120

Cliente(idPessoa→Pessoa, idEstatuto→Estatuto)

- Número de identificação (idPessoa) é **foreign key** para Pessoa, classe mãe de Cliente.
- Herda a totalidade das restrições relativas à relação Pessoa.
- idEstatuto funciona como **foreign key** para Estatuto e é **not null**.

Estatuto(idEstatuto, desconto, tipo)

- A cada estatuto está associado um desconto e um tipo, bem como idEstatuto, sendo todos estes **not null**.
- idEstatuto funciona enquanto **primary key** da relação.

- Tipo deve igualar a um dos tipos pré-definidos (estudante, sénior, juvenil, regular).
- desconto deverá ser maior ou igual a 0

Funcionario(idPessoa→Pessoa, ID, salario, horas_semanais, idCinema→Cinema, idBilheteira→Bilheteira)

- Herda a totalidade das restrições relativas à relação Pessoa.
- Não existem dois ID's únicos, logo ID é um atributo **unique**.
- idCinema funciona como **foreign key** para o cinema no qual o funcionário em questão está empregado.
- idBilheteira é **foreign key** para a bilheteira associada ao funcionário.
- salario e horas_semanais deverão ser maior que 0.
- a idade de um funcionário terá de ser superior a 18.

Bilheteira(idBilheteira, montante, idCinema→Cinema)

- todos os atributos são **não nulos**.
- idBilheteira representa uma **primary key**
- idCinema representa uma **foreign key** para o Cinema correspondente a essa mesma bilheteira.
- montante deve ser igual ou superior a 0.

Snack(idSnack, nome, tamanho, preco, idMenu→Menu)

- idSnack e nome são atributos **não nulos**, enquanto tamanho, preço e idMenu poderão ser nulos.
- idSnack funciona como **primary key** da relação.
- idMenu representa uma **foreign key** para o Menu ao qual pertence esse Snack.
- Tamanho deve corresponder a um dos tamanhos pré-definidos (pequeno, médio, grande ou não definido).
- É verificado se, caso o snack esteja associado a um Menu, este tem preço e tamanho a nulo, ou caso não esteja associado a um menu, o preço e o tamanho são não nulos.
- Preço deve ser superior a 0.

Menu(idMenu, tamanho, preco)

- Todos os atributos são **não nulos**.
- idMenu é **primary key** da relação.
- Tamanho deve corresponder a um dos tamanhos pré-definidos (pequeno, médio ou grande).
- Preço deve ser superior a 0.

Bilhete(idBilhete, preco, lugar, idPessoa→Cliente, idSessao→Sessao)

- Todos os atributos são **não nulos**.
- idBilhete é **primary key** da relação.
- idSessao representa uma **foreign key** para uma Sessão.
- idPessoa representa uma **foreign key** para um Cliente (consumidor do bilhete).
- Preço deve ser superior a 0.

Sessao(idSessao, hora_inicio, hora_fim, data, idFilme→Filme, idSala→Sala, lugares_livres)

- Todos os atributos são **não nulos**.
- idSessao é **primary key** da relação.
- idFilme representa uma **foreign key** para o Filme associado à sessão.
- idSala representa uma **foreign key** para a sala de exibição da sessão em questão.

Sala(idSala, numero, capacidade, tipo, idTipoSala→TipoSala, idCinema→Cinema)

- Todos os atributos são **não nulos**.
- idSala é **primary key** da relação.
- idTipoSala representa uma **foreign key** para o tipo de sala (TipoSala) em questão.
- idCinema representa uma **foreign key** para o cinema onde a Sala se encontra.
- capacidade deverá ser maior que 0

TipoSala(idTipoSala, precoSala, tipo)

- Todos os atributos são **não nulos**.
- idTipoSala é **primary key** da relação.
- Tipo deve corresponder a um dos tipos pré definidos (regular, 3D, IMAX, X Vision).
- Preço da sala deve ser superior a 0.

Filme(idFilme, nome, genero, duracao, idade_minima, sinopse, classificacao)

- Todos os atributos são **não nulos**.
- idFilme é **primary key** da relação.
- genero deve corresponder a um ou mais géneros pré definidos (ação, animação, aventura, comédia, documentário, drama, fantasia, ficção científica, histórico, musical, policial, romance, suspense, terror ou western).
- Duracao terá de ser maior que 0.
- Idade_minima deve estar compreendida entre 3 e 18.
- classificacao deve estar compreendida entre 0 e 10.

CinemaFilme(idFilme→Filme, idCinema→Cinema, data_estreia, data_fim)

- Todos os atributos são **não nulos**.
- idFilme e idCinema funcionam como **primary key** da relação.
- idCinema representa uma **foreign key** para um Cinema.
- idFilme representa uma **foreign key** para um Filme.
- data_fim deve ser superior a data_estreia.

Fatura(idFatura, data, total, idBilheteira→Bilheteira, idPessoa→Cliente)

- Todos os atributos são **não nulos**.
- idFatura funciona como **primary key** da relação.
- idBilheteira representa uma **foreign key** para Bilheteira.
- idPessoa representa uma **foreign key** para o cliente ao qual pertence a fatura.
- total terá de ser maior que 0

LinhaFatura(idLinhaFatura, quantidade, idSnack→Snack, idMenu→Menu, idFatura→Fatura)

- quantidade e idFatura são **não nulos**
- idSnack e idMenu poderão ser nulos (se idSnack for nulo, idMenu terá de ser não nulo e vice-versa)
- idLinhaFatura funciona como **primary key** da relação.
- idSnack representa uma **foreign key** para Snack
- idMenu representa uma **foreign key** para Menu
- idFatura representa uma **foreign key** para Fatura
- quantidade deve ser maior que 0

BilheteiraMenu(idBilheteira→Bilheteira, idMenu→Menu)

- Todos os atributos são **não nulos**.
- idBilheteira é **primary key** da relação.
- idBilheteira representa uma **foreign key** para uma Bilheteira.
- idMenu representa uma **foreign key** para um Menu.

BilheteiraSnack(idBilheteira→Bilheteira, idSnack→Snack)

- Todos os atributos são **não nulos**.
- idBilheteira é **primary key** da relação.
- idBilheteira representa uma **foreign key** para uma Bilheteira.
- idSnack representa uma **foreign key** para um Snack.

BilheteiraBilhete(idBilheteira→Bilheteira, idBilhete→Bilhete)

- Todos os atributos são **não nulos**.
- idBilheteira e idBilhete são **primary key** da relação.
- idBilheteira representa uma **foreign key** para uma Bilheteira.
- idBilhete representa uma **foreign key** para um Bilhete.

Recorreu-se ao uso de restrições na produção das diversas classes de modo a garantir uma base de dados mais fiável e eficiente, fornecendo uma segurança

adicional ao utilizador. Entre as quais, destacam-se **primary key**, **foreign key**, **not null**, **unique** e **check**. Certas restrições, relacionadas diretamente com Triggers, não foram ainda implementadas, uma vez que não foram ainda abordadas nas aulas.

No caso da **primary key**, esta visa identificar os atributos a partir dos quais, posteriormente, será dado acesso a outro(s) atributo(s) de uma determinada relação. Geralmente, estas são constituídas por um identificador único construído propositadamente para esta função, de forma a facilitar a manutenção da base de dados, tornando-a mais independente em relação a estes dados e, conseqüentemente, melhorando a sua eficiência.

No caso das **foreign keys**, estas permitem a comunicação entre diferentes classes, aumentando assim a integridade referencial. Desta forma, é reforçada qualquer ligação que uma classe possua com outra.

Relativamente à restrição **check**, esta visa confirmar a validade dos dados introduzidos, verificando se estes cumprem determinada condição, prevenindo, assim, a introdução de dados incompatíveis com contexto em questão, assegurando a fiabilidade da mesma.

Recorreu-se ainda à restrição **unique**, de modo a prevenir ambiguidade entre certos atributos não chave de uma mesma relação. Um exemplo desta restrição diz respeito ao número do cartão de cidadão, que, apesar de não constituir um atributo chave, não deve ser permitida a existência de dois com o mesmo valor.

Por fim, **not null** foi aplicado a grande parte dos atributos utilizados, uma vez que assegura a existência dos mesmos com valores associados aquando da criação de cada relação. Não obstante, em certos atributos, não foi necessária a aplicação desta restrição, como é o caso do idMenu de um Snack, uma vez que nem todos os Snacks terão de estar obrigatoriamente associados a um Menu.

6 Interrogação da Base de Dados

1. Qual o preço final que cada cliente pagou pelo bilhete?
2. Qual o filme mais visto (e respetiva quantidade de bilhetes vendidos) em cada cinema?
3. Qual o intervalo de tempo médio (em dias) em que um cliente vê um filme na cadeia de cinemas?
4. Quais os filmes que ainda não venderam bilhetes?
5. Quais os cinemas em que o número de clientes (mensais) foi superior à média de clientes mensais na cadeia de cinemas?
6. Qual o género mais visto e respetivo número de bilhetes vendidos?
7. Qual o intervalo de exibição mínimo e máximo dos filmes da cadeia de cinemas?
8. Qual é a classificação média dos 3 filmes mais vistos?
9. Tendo em conta a receita de cada filme, qual a percentagem desta é da data de estreia?
10. Qual a Faturação mensal? É considerado a receita obtida com a venda dos bilhetes, snacks e menus.

7 Adição de Gatilhos à Base de Dados

1. Controla número de vagas das sessões (elimina uma vaga por cada bilhete e controla a capacidade das sessões);
2. Verifica se o cliente tem idade mínima para assistir ao filme do bilhete que pretende comprar;
3. Ao inserir um bilhete, o preço deste é automaticamente atualizado consoante o estatuto do cliente e o tipo de sala.