

Cinema

Estudantes:

- + André Costa - up201905916
- + David Ferreira - up202006302
- + Guilherme Carvalho - up202005369

Professores:

- + Michel Celestino Paiva Ferreira
- + Carla Alexandra Teixeira Lopes
- + Lázaro Gabriel Barros da Costa

Tema do trabalho:

- + Cinema

Índice

1. Capa	01
2. Índice	02
3. Descrição	03
4. Diagrama Conceptual - UML	04
5. Diagrama Conceptual - UML revisto	05
6. Modelo Relacional	06
7. Dependências Funcionais e Formas Normais	07-08
8. Restrições	09-11

Descrição

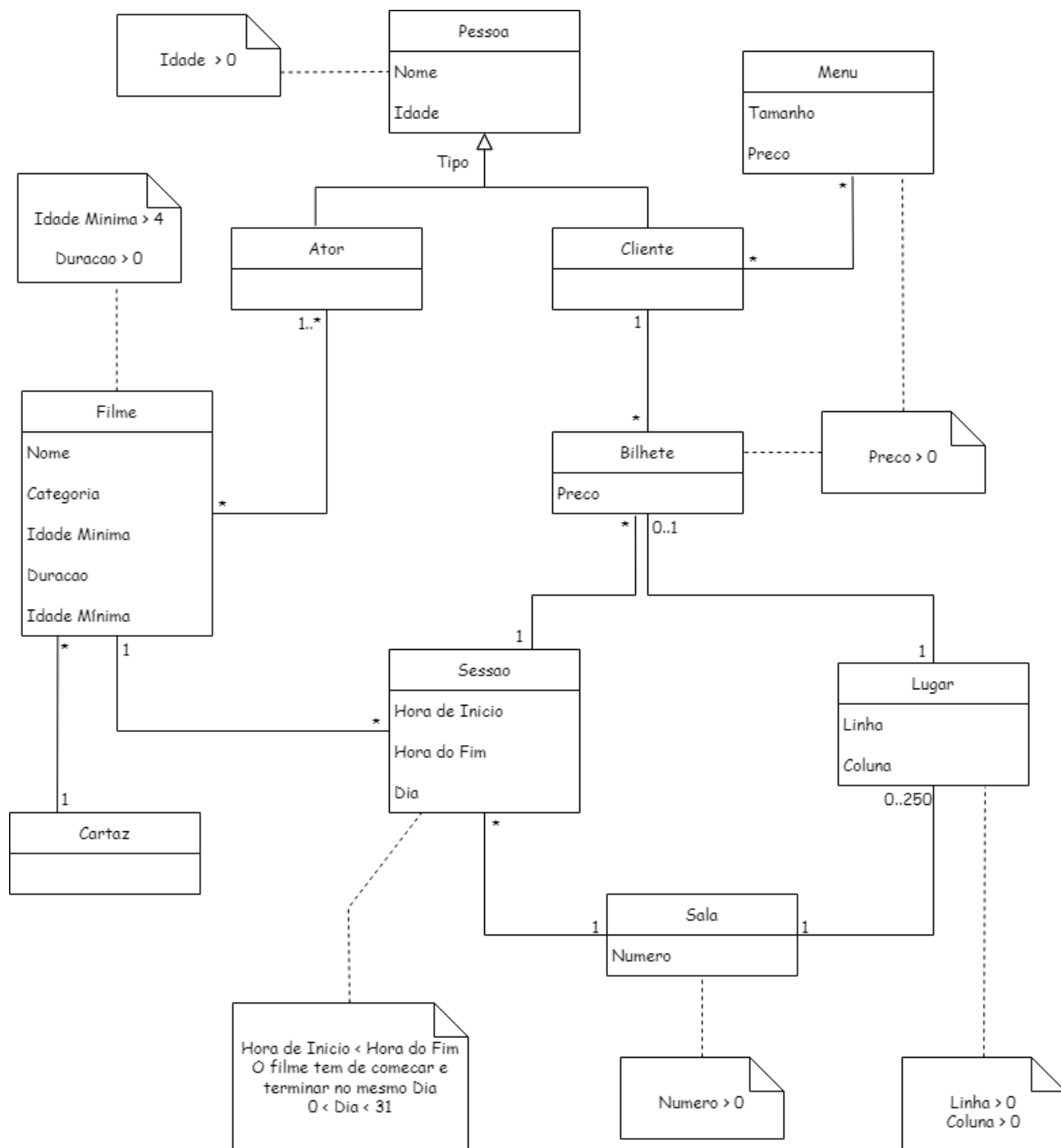
Pretende-se guardar informação relativa ao funcionamento dos Cinema UCI. O Cinema tem a seu cargo a manutenção do cartaz, com a CapaUCI e um IdCartaz, os filmes disponíveis e a respetiva informação de cada filme, e os clientes.

De cada cliente interessa saber que filme gostariam de ver e a sessão correspondente (IdSessao, hora-início, hora-final e o dia). Estes ainda têm à sua disposição um menu consumível, que podem ter diversos tamanhos, variando assim os seus preços.

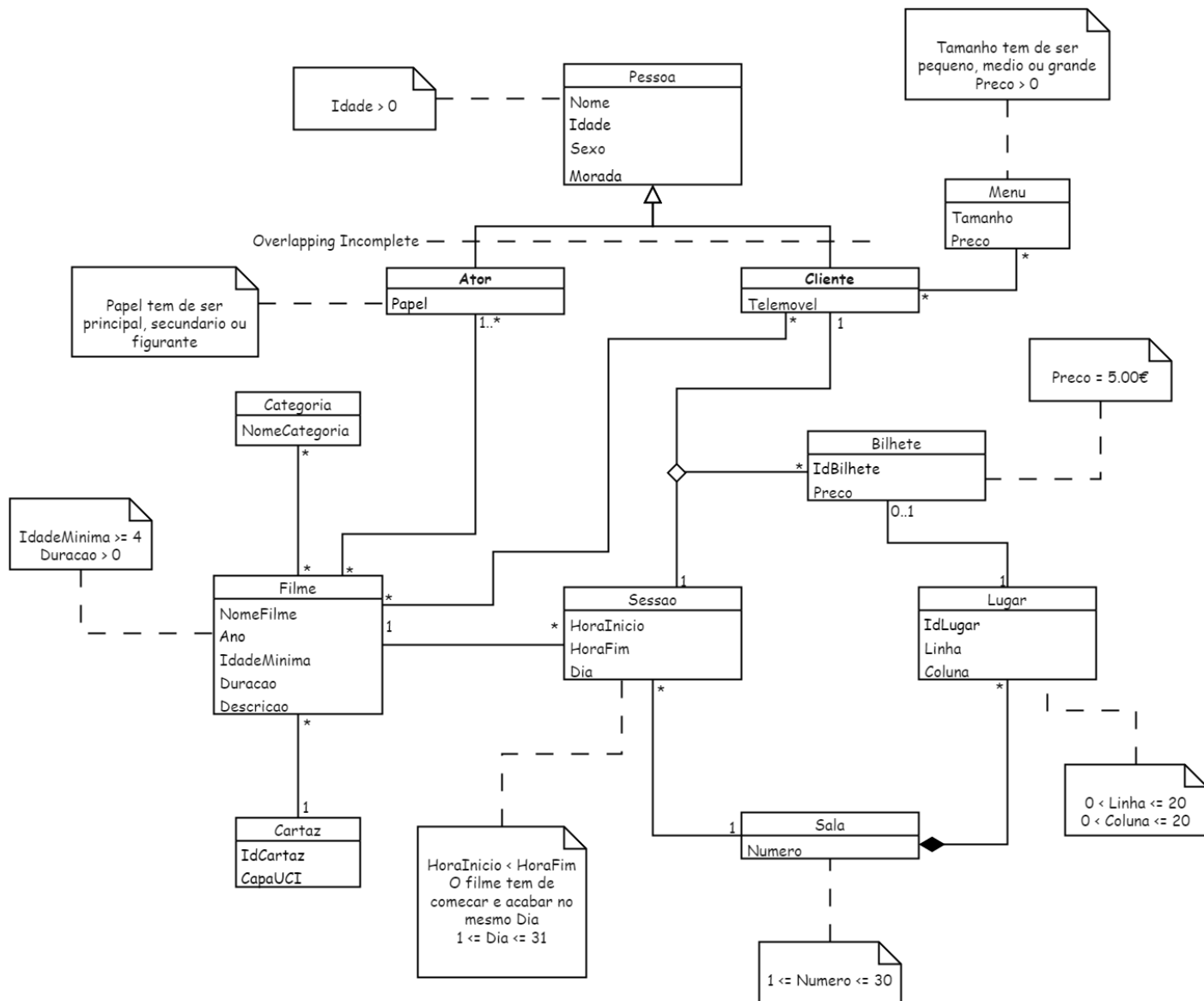
Para ajudar os clientes na escolha dos filmes, estes para além do nome e da sua descrição, têm ainda as várias categorias, sessões disponíveis (com o respetivo número da sala atribuída), a duração, a idade mínima e os atores envolvidos. Os atores e os clientes, são ambos pessoas, descritos pelo nome, idade, sexo e morada. Dos atores ainda se sabe se fazem o papel principal, secundário ou de figurantes.

Para o cliente assistir a um filme, tem de comprar o bilhete com um IdBilhete e um preço associado, cujo tem a informação do lugar (IdLugar, linha, coluna), do número da sala e da sessão correspondente. Tem que deixar também o seu número de telemóvel, para caso seja necessário ser contactado.

Modelo Conceptual - UML



Modelo Conceptual – UML revisto



Modelo Relacional

- Cartaz(IdCartaz, CapaUCI)
- Filme(NomeFilme, Ano, IdadeMinima, Duracao, Descricao, IdCartaz→Cartaz)
- Categoria(NomeCategoria)
- Sala(Numero)
- Sessao(HoraInicio, HoraFim, Dia, Numero→Sala, NomeFilme→Filme, Ano→Filme)
- Lugar(IdLugar, Linha, Coluna, Numero→Sala)
- Pessoa(Nome, Idade, Morada, Sexo)
- Ator(Nome→Pessoa, Idade→Pessoa, Morada→Pessoa, Papel)
- Cliente(Nome→Pessoa, Idade→Pessoa, Morada→Pessoa, Telemovel)
- Bilhete(IdBilhete, Preco, {HoraInicio, HoraFim, Dia, Numero}→Sessao, IdLugar→Lugar)
- Menu(Tamanho, Preco)

Many-to-Many Associations:

- FilmeCategoria(NomeFilme→Filme, Ano→Filme, NomeCategoria→Categoria)
- FilmeAtor(NomeFilme→Filme, Ano→Filme, {Nome, Idade, Morada}→Ator)
- FilmeCliente(NomeFilme→Filme, Ano→Filme, {Nome, Idade, Morada, Telemovel}→Cliente)
- ClienteMenu({Nome, Idade, Morada, Telemovel}→Cliente, Tamanho→Menu)

Dependências Funcionais e Formas Normais

Cartaz(IdCartaz, CapaUCI)

IdCartaz → CapaUCI

Filme(NomeFilme, Ano, IdadeMinima, Duracao, Descricao, IdCartaz → Cartaz)

{NomeFilme, Ano} → {IdadeMinima, Descricao, Duracao, Id}

Sessao(HoraInicio, HoraFim, Dia, Numero → Sala, NomeFilme → Filme, Ano → Filme)

{HoraInicio, HoraFim, Dia, Numero} → {NomeFilme, Ano}

Lugar(IdLugar, Linha, Coluna, Numero → Sala)

IdLugar → {Linha, Coluna, Numero}

Pessoa(Nome, Idade, Morada, Sexo)

{Nome, Idade, Morada} → Sexo

Ator(Nome → Pessoa, Idade → Pessoa, Morada → Pessoa, Papel)

{Nome, Idade, Morada} → Papel

Cliente(Nome → Pessoa, Idade → Pessoa, Morada → Pessoa, Telemovel)

Telemovel → {Morada, Nome, Idade}

Bilhete(IdBilhete, Preço, {HoraInicio, HoraFim, Dia, Numero}->Sessao, IdLugar->Lugar)

IdBilhete->{Preço, {HoraInicio, HoraFim, Dia, Numero}, IdLugar}

Menu(Tamanho, Preço)

Tamanho->Preço

FilmeCategoria(NomeFilme->Filme, Ano->Filme, NomeCategoria->Categoria)

Não tem;

FilmeAtor(NomeFilme->Filme, Ano->Filme, {Nome, Idade, Morada}->Ator)

Não tem;

FilmeCliente(NomeFilme->Filme, Ano->Filme, {Nome, Idade, Morada, Telemovel}->Cliente)

Não tem;

ClienteMenu({Nome, Idade, Morada, Telemovel}->Cliente, Tamanho->Menu)

Não tem;

As nossas relações estão todas na BCNF, pois ou as dependências funcionais são triviais, apesar de estas não estarem representadas, tal como Numero->Numero da relação Sala, ou os atributos do lado esquerdo são chaves/superchaves.

As nossas relações estão todas na 3NF, pois para cada dependência funcional não-trivial, os atributos do lado esquerdo são chaves/superchaves ou os atributos do lado direito são todos primos.

Restrições

Cartaz(IdCartaz, CapaUCT)

O Id auto-incrementa-se.

PRIMARY KEY(IdCartaz)

Filme(NomeFilme, Ano, IdadeMinima, Duracao, Descricao, IdCartaz->Cartaz)

Uma pessoa para ir assistir a um filme tem de ter no mínimo 4 anos, apesar que pode variar de filme para filme.

Um filme tem de ter mais do que 0 mins;

PRIMARY KEY(NomeFilme, Ano)

CHECK(Duracao > 0)

CHECK(IdadeMinima > 4)

Categoria(NomeCategoria)

PRIMARY KEY(NomeCategoria)

Sala(Numero)

Só existem 30 salas no cinema numeradas a começar no número 1.

PRIMARY KEY(Numero)

CHECK(Numero >= 1 AND Numero <= 30)

Sessao(HoraInicio, HoraFim, Dia, Numero->Sala, NomeFilme->Filme, Ano->Filme)

A hora do fim de um filme é sempre maior do que a hora do inicio.

O filme tem de começar e acabar no mesmo dia. E o dia tem de pertencer aos dias do mês.

(Generalizamos os meses para meses com 31 dias)

PRIMARY KEY(HoraInicio, HoraFim, Dia, Numero)

CHECK(HoraInicio < HoraFim)

CHECK(HoraFim <= '23:59')

CHECK(Dia > 0 AND Dia <= 31)

Lugar(IdLugar, Linha, Coluna, Numero->Sala)

O Id auto incrementa-se. O número das linhas e das colunas estão numerados de 0 a 20.

PRIMARY KEY(IdLugar)

CHECK(Linha > 0 AND Linha <= 20)

CHECK(Coluna > 0 AND Coluna <= 20)

Pessoa(Nome, Idade, Morada, Sexo)

Qualquer pessoa tem mais do que 0 anos.

PRIMARY KEY(Nome, Idade, Morada)

CHECK(Idade > 0)

CHECK(Sexo = 'M' OR Sexo = 'F')

Ator(Nome->Pessoa, Idade->Pessoa, Morada->Pessoa, Papel)

Um ator ou tem papel principal, papel secundário ou faz de figurante.

PRIMARY KEY(Nome, Idade, Morada)

CHECK(Papel = 'Papel Principal' OR Papel = 'Papel Secundario' OR Papel = 'Figurante')

Cliente(Nome->Pessoa, Idade->Pessoa, Morada->Pessoa, Telemovel)

O número de telemóvel é único, não existem 2 iguais e têm todos 9 números.

PRIMARY KEY(Nome, Idade, Morada, Telemovel)

UNIQUE(Telemovel)

Bilhete(IdBilhete, Preco, {HoraInicio, HoraFim, Dia, Numero}-> Sessao, IdLugar->Lugar)

O Id auto incrementa-se. Os bilhetes não são grátis.

PRIMARY KEY(IdBilhete)

CHECK(Preco = '5.00€')

Menu(Tamanho, Preco)

Um cliente tem acesso a menus com 3 tamanhos distintos, pequeno, médio ou grande.

E os menus não são grátis.

PRIMARY KEY(Tamanho)

CHECK(Tamanho = 'Pequeno' OR Tamanho = 'Medio' OR Tamanho = 'Grande')

CHECK(Preco > '3.00€')

FilmeCategoria(NomeFilme->Filme, Ano->Filme, NomeCategoria->Categoria)

PRIMARY KEY(NomeFilme, Ano, NomeCategoria)

FilmeAtor(NomeFilme->Filme, Ano->Filme, {Morada, Nome, Idade}->Ator)

PRIMARY KEY(NomeFilme, Ano, {Morada, Nome, Idade})

FilmeCliente(NomeFilme->Filme, Ano->Filme, {Morada, Nome, Idade, Telemovel}->Cliente)

PRIMARY KEY(NomeFilme, Ano, {Morada, Nome, Idade, Telemovel})

ClienteMenu({Nome, Idade, Morada, Telemovel}->Cliente, Tamanho->Menu)

PRIMARY KEY({Morada, Nome, Idade, Telemovel}, Tamanho)