



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2017/2018

Bilheteira de Cinema

Diogo Emanuel da Silva Nogueira (a78957)

Diogo Francisco Araújo Leite da Costa (a78485)

Mariana Lino Lopes Costa (a78824)

Sarah Tifany da Silva (a76867)

Novembro, 2017

BD

Data de Receção	
Responsável	
Avaliação	
Observações	

Bilheteira de Cinema

Diogo Emanuel da Silva Nogueira (a78957)

Diogo Francisco Araújo Leite da Costa (a78485)

Mariana Lino Lopes Costa (a78824)

Sarah Tifany da Silva (a76867)

Novembro, 2017

Resumo

Este projeto consiste em desenvolver uma BD que possa dar suporte a um Sistema que permita o correto funcionamento de uma Bilheteira de Cinema, auxiliando a fluidez e boa assistência a nível de serviço, desde a compra do Bilhete até à gestão logística e financeira da mesma.

Área de Aplicação: Desenho e Administração de BD para uma Bilheteira de um Cinema.

Palavras-Chave: Cinema, Bilheteira, Bilhetes, Entretenimento, Simplicidade, Organização, Rapidez, Descomplicação, Fluidez, Assistência, BD, Entidades, Atributos.

Índice

1. Definição do Sistema	1
1.1. Contexto de aplicação do Sistema	1
1.2. Fundamentação da implementação da BD	2
1.3. Análise da viabilidade do processo	3
2. Levantamento e Análise de Requisitos	4
2.1. Método de levantamento e análises de requisitos adotado	4
2.1.1. Requisitos de descrição	4
2.1.2. Requisitos de exploração	5
2.1.3. Requisitos de controlo	5
2.2. Análise geral dos requisitos	6
3. Modelação Conceptual	7
3.1. Apresentação da abordagem de modelação realizada	7
3.2. Identificação e caracterização das Entidades	7
3.3. Identificação e caracterização dos Relacionamentos	8
3.4. Identificação e caracterização das associações dos Atributos com as Entidades	9
3.5. Detalhe ou generalização de Entidades	12
3.6. Apresentação e explicação do Diagrama ER	13
3.7. Validação do modelo de dados com o utilizador	15
4. Modelação Conceptual	16
4.1. Construção e validação do modelo de dados lógico	16
4.2. Desenho do Modelo Lógico	18
4.3. Validação do modelo através da normalização	19
4.4. Validação do modelo com as interrogações do utilizador	20
4.5. Validação do modelo com as transações estabelecidas	22
4.6. Revisão do modelo lógico com o utilizador	23
5. Implementação Física	24
5.1. Seleção do Sistema de gestão de BD	24
5.2. Tradução do esquema lógico para o Sistema de gestão de bases de dados escolhido em SQL	25
5.3. Tradução das interrogações do utilizador para SQL	27
5.4. Tradução das transações estabelecidas para SQL(alguns exemplos)	28
5.5. Escolha, definição e caracterização de índices em SQL	29
5.6. Estimativa do espaço em disco da BD e taxa de crescimento anual	29
5.7. Definição e caracterização dos mecanismos de segurança em SQL	31
5.8. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	32
5.9. Revisão do Sistema implementado pelo utilizador	32
6. Conclusões e trabalho futuro	33
7. Referências Bibliográficas	34
8. Lista de Siglas e Acrónimos	35

Índice de Figuras

Figura 1: Diagrama ER (Modelo Concetual)	13
Figura 2: Modelo Lógico	18

Índice de Tabelas

Tabela 1: Caracterização das Entidades	7
Tabela 2 : Caracterização dos Relacionamentos entre Entidades	8
Tabela 3 :Caracterização dos Atributos da Entidade Cliente	9
Tabela 4 :Caracterização dos Atributos da Entidade Bilhete	10
Tabela 5 :Caracterização dos Atributos da Entidade InfoBilhete	10
Tabela 6 :Caracterização dos Atributos da Entidade Bilhete	11
Tabela 7 :Tamanho que cada tabela ocupa em KB	30

1. Definição do Sistema

Neste capítulo inicial é feita, de forma repartida, uma pequena apresentação acerca de todo o nosso trabalho. Numa fase inicial, contextualizamos o problema a resolver refletindo a ideia do porquê de existir a necessidade do desenvolvimento da nossa BD. De seguinte, demonstramos uma fundamentação para a qual indica a necessidade dum Sistema de BD para a empresa-Cliente. Por fim, neste capítulo, iremos analisar toda a viabilidade do processo de criação e manutenção da SBD.

1.1. Contexto de aplicação do Sistema

O Cinema é considerado a técnica e a arte de fixar e reproduzir imagens que suscitam a impressão de movimento. As obras Cinematográficas, mais conhecidas como Filmes, são produzidas através das gravações de imagens do mundo com câmaras adequadas, ou pela sua criação utilizando técnicas de animação ou efeitos visuais (Cinematography, 2017). Desde 1895, o Ano em que foi lançado o primeiro Filme no Cinema, que esta área tem ganho cada vez mais aderentes e crescido exponencialmente, não só pela evolução da tecnologia permitindo a criação de Filmes mais grandiosos e com uma maior qualidade, mas também pela maior procura e por se ter tornado uma excelente forma de entretenimento.

O Cinema *PopCorn*, inaugurado dia 25 de Maio de 2016, situa-se nas proximidades do centro de Braga, mais concretamente, no centro comercial *Minho Center*, possuindo uma vasta programação de Filmes e horários.

A Bilheteira para a venda dos Bilhetes é constituída por 6 Caixas e encontra-se aberta de segunda a sábado, do meio dia até à uma da manhã, e ainda aos domingos das dez da manhã até à meia noite.

O Cliente escolhe o Filme através de todo cartaz disponível e exposto na Bilheteira, onde se encontram informações como o título do Filme, Género, Duração, data de estreia e por vezes, até mesmo o País, Elenco e Realizador do mesmo. De seguida, selecciona a sessão que, obviamente, varia e é definida consoante a Duração de cada Filme.

Os Bilhetes podem ser adquiridos para o próprio dia ou seguintes e consequentemente para qualquer Sessão e dia da semana.

O *PopCorn* conta com 12 Salas, que têm desde 100 até 400 Lugares sendo que no total o Cinema possui a capacidade de mais de 2500 Lugares. Nas Salas do *PopCorn* os espectadores encontram qualidade de som e imagem, com as mais recentes tecnologias existentes, como por exemplo, o Sistema de som Dolby Atmos.

Cada Filme está associado a uma Sala contendo as suas respetivas sessões. Os Lugares de cada Sala são identificados por uma letra que indica a fila e um número que designa a cadeira.

Existem Lugares com cadeiras de assento normais, *chaise longs* e até mesmo cadeiras especiais para casais, todas estas caracterizadas pela sua comodidade, de modo a que todos os Clientes possam usufrir do Filme com o maior conforto possível.

Dentro de todo o estabelecimento existe ainda um bar onde cada Cliente pode comprar as suas pipocas, os mais diversos refrigerantes, chocolates, gomas e até mesmo sushi.

1.2. Fundamentação da implementação da BD

Com um maior fluxo de dados criados numa Bilheteira de Cinema, desde a criação de diversos Bilhetes, com as suas sessões, Salas e preços associados, existe a necessidade de se criar um Sistema fiável e funcional de forma a interligar e gerir todo o processo da compra do Bilhete até à gestão final de preços, lucros e estatísticas associadas às finanças, mas também o tratamento de dados internos da Bilheteira de forma a melhorar e modificar os serviços internos da mesma.

Em consequência, a modelação e implementação dum Sistema de BD simplifica este processo a curto, médio e longo prazo de vida da Bilheteira.

1.3. Análise da viabilidade do processo

Com a fundamentação delineada, o processo tem de ser criado para tornar toda a Bilheteira mais viável. Ao informatizar toda a informação dos Filmes, tal como Nome, Elenco, Duração, Género, mas também a informação relativa à estrutura física do estabelecimento como o nº de Salas, sessões e Lugares, existe uma facilidade para manipular e observar todos estes dados.

Filtrando e analisando estas informações “base” acabamos por obter conhecimentos derivados das mesmas de forma a melhorar e viabilizar a Bilheteira, tal como, analisar sessões consoante a Duração dos Filmes ou até prever o nº de vendas de Bilhetes consoante o Género do Filme e dessa forma escolher/adaptar as Salas escolhidas para o mesmo. Apenas estes dois exemplos de derivações da informação “crua” tornam o processo de adoção dum SBD algo não só apelativo como necessário e viável.

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e análises de requisitos adotado

Um dos pontos mais fulcrais para um correto funcionamento de qualquer BD passa pelo levantamento e respetiva análise de requisitos. Dessa forma, neste capítulo, vamos precisamente examinar todos os requisitos que a nossa BD deve atender, fazendo uma divisão metódica e organizada dos mesmos.

2.1.1. Requisitos de descrição

Em consequência à entrevista ao nosso empregador, foram apurados os seguintes requisitos de descrição que são considerados imprescindíveis para uma correta ligação entre o Cliente e a sua respetiva compra.

RD1 - Cada Bilhete tem relacionado o seu respetivo Cliente/comprador e Filme desejado. No Bilhete devem estar registados as informações básicas/essenciais de forma a permitir uma correta gestão de todos os Bilhetes extraídos nas demais Caixas da Bilheteira. Pretende-se, portanto, que:

- Cada Cliente tenha a possibilidade de adquirir o número de Bilhetes que desejar, sendo cada Bilhete dirigido apenas a um Lugar na Sala e Sessão;
- Um Filme esteja associado aos vários Bilhetes comprados pelos Clientes para o mesmo;
- Cada Bilhete tenha conectado a si as informações correspondentes ao Lugar bem como a Sessão e, consequentemente, a Sala onde o Filme vai decorrer.

RD2 - Cada Cliente detém a possibilidade de comprar vários Bilhetes. É identificado pelo seu NIF e deve facultar o seu Nome, o seu Contacto e ainda, caso desejado, a sua Data de Nascimento. Desta forma:

- Cada Bilhete pertence apenas a um Cliente.

RD3 - Cada Filme deve conter as informações consideradas fundamentais para a exposição ao Cliente. Considera-se obrigatório para esse fim a divulgação do seu Nome, Duração e Ano de estreia. Já o País, o Realizador, o Elenco e ainda a Sinopse são esclarecimentos facultativos e que podem por isso, não serem mostrados ao par dos outros.

- Cada Bilhete diz respeito apenas a um Filme. Mesmo assim, não é exequível o mesmo Bilhete possuir várias sessões ainda que para o mesmo Filme.

2.1.2. Requisitos de exploração

Consoante o pedido do nosso empregador, a nossa BD, além de alguns requisitos base de exploração, pede-se também que seja possível nos fornecer a seguinte informação:

- Saber a receita total de um certo Filme;
- Saber a faixa etária mais comum para um determinado Filme;
- Calcular qual o Género de Filme que gera mais receitas ao Cinema;
- Conhecer qual a Sessão mais visitada/visionada de um certo Filme ou até mesmo do próprio Cinema;
- Obter uma tabela das compras de Bilhetes de um certo Cliente;
- Saber o Género de Filme mais popular de um Cliente.

2.1.3. Requisitos de controlo

Pretendemos, com estes requisitos, explicitar todas as possibilidades que o utilizador e ainda o funcionário presente na Caixa de compra de Bilhetes, podem manipular/rever. Perante a nossa BD e também perante todos os requisitos previamente definidos pela necessidade imposta pelo empregador, podemos estabelecer os seguintes requisitos de controlo:

- **Por parte do Funcionário de Caixa, existe a hipótese de:**
 - Inserir, atualizar e eliminar dados relativos ao Cliente, como Contacto, Nome e Data de Nascimento;
 - Inserir e atualizar a informação referente ao Filme, Sessão e também Lugar escolhido pelo Cliente;
 - Consultar tudo o que diz respeito ao Bilhete em si de modo a atender ao pedido do Cliente da forma mais correta possível.
- **Por parte da Administração do Cinema, existe a possibilidade de:**
 - Efetuar todas as operações exequíveis por parte do Funcionário de Caixa;
 - Tem como possibilidade extra, aceder a quaisquer dados existentes em toda a BD, ou seja, todas as informações do Cliente, dos vários Filmes expostos e ainda dos rendimentos da Bilheteira;
 - E ainda criar, alterar ou destruir dados relativos a todas as Entidades.

2.2. Análise geral dos requisitos

Fazendo uma análise geral deste levantamento de requisitos, podemos manifestar alguns comentários que consideramos relevantes. Primeiro de tudo, não dispensar o peso e importância desta análise pois, com ela, podemos desde o começo estipular a maneira correta de como deve a nossa BD comportar-se. Além disso, não só aprofundamos o conhecimento acerca do funcionamento da nossa BD como também fomos capazes e nos vimos, em algumas situações, obrigados a corrigir erros que não faziam de todo sentido e que comprometiam posteriormente a viabilidade de todo o processo.

Assim conseguimos assegurar que todas as relações entre os requisitos e as suas respectivas propriedades que definimos se relacionam corretamente e, acima de tudo, dão resposta ao que o empregador definiu como crux para a implementação da nossa BD.

3. Modelação Conceptual

3.1. Apresentação da abordagem de modelação realizada

Para o correto desenvolvimento de uma BD é necessário, numa fase inicial, conceber o modelo conceptual. Para isso, recorreremos a um diagrama ER de forma a o podermos representar. Este diagrama é composto por Entidades que são representativas de uma classe de objetos sobre os quais se pretende armazenar a informação, Atributos que apresentam a descrição das Entidades bem como os relacionamentos que representam as associações relevantes entre as várias Entidades.

3.2. Identificação e caracterização das Entidades

Depois da análise do problema em questão, foi-nos possível identificar 4 Entidades. Estas Entidades encontram-se estabelecidas e caracterizadas na seguinte tabela.

Entidade	Descrição	Ocorrência
Bilhete	Termo geral que descreve o Bilhete vendido pela Bilheteira do Cinema.	Um Bilhete pode ser comprado unicamente por um Cliente e pertence apenas a um Filme e uma Sala.
Cliente	Termo geral que descreve o individuo que compra o Bilhete de Cinema.	Um Cliente pode comprar 1 ou mais Bilhetes para um determinado Filme ou até mesmo para vários, se assim o pretender.
Filme	Termo geral que descreve os Filmes que estão em exibição no Cinema.	A Bilheteira vende vários Bilhetes dos Filmes que estão em exibição. Um determinado Filme pode estar em exibição numa ou em mais Salas.
InfoBilhete	Termo geral que possui as informações básicas da do Bilhete em questão.	Um Bilhete está associado apenas a um InfoBilhete.

Tabela 1: Caracterização das Entidades

3.3. Identificação e caracterização dos Relacionamentos

Após termos procedido à identificação e caracterização de todas as Entidades presentes na BD, iremos realizar o mesmo processo para as relações das mesmas. Deste modo, vamos classificar cada relacionamento pelas suas respetivas Entidades, pela cardinalidade que possuem e por uma breve descrição.

Entidade	Relacionamento	Entidade	Cardinalidade	Descrição
Cliente	Compra	Bilhete	1:N	Um Cliente pode comprar um ou mais Bilhetes.
Bilhete	É comprado	Cliente	1:1	Um Bilhete só pode ser comprado por um Cliente.
Filme	Tem	Bilhete	1: N	Um Filme pode ter um ou mais Bilhetes à venda.
Bilhete	Está associado	Filme	1:1	Um Bilhete está associado a um Filme.
Bilhete	Está associado	InfoBilhete	1:1	Um Bilhete tem a sua respetiva informação.

Tabela 2 : Caracterização dos Relacionamentos entre Entidades

3.4. Identificação e caracterização das associações dos Atributos com as Entidades

Em consequência da demonstração atrás definida das Entidades e Relacionamentos respetivamente, iremos identificar e caracterizar os Atributos associados às mesmas.

Entidade: Cliente				
Atributo	Descrição	Tipo de Dados e Domínio	Nulo	Tipo de Atributo
NIF	Número de Identificação Fiscal do Cliente	Número inteiro positivo até 9 algarismos	Não	Identificador (Simples)
Nome	Nome do Cliente	String variável até 45 carateres	Não	Simples
Data de Nascimento	Data de Nascimento do Cliente	DATETIME	Sim	Simples
Contacto	Número telefónico do Cliente	Número inteiro positivo até 9 algarismos	Não	Simples

Tabela 3 :Caracterização dos Atributos da Entidade Cliente

Entidade: Bilhete				
Atributo	Descrição	Tipo de Dados e Domínio	Nulo	Tipo de Atributo
DataCompra	Data correspondente ao momento da compra (até à fração de segundo)	DATETIME	Não	Identificador (Simples)
Preço	Preço que diz respeito ao Bilhete	Decimal de 5 algarismos com 2 fracionários	Não	Simples
Nº de Caixa	Número da Caixa da compra	Número inteiro positivo com 1 algarismo	Sim	Simples

Tabela 4 :Caracterização dos Atributos da Entidade Bilhete

Entidade: InfoBilhete				
Atributo	Descrição	Tipo de Dados e Domínio	Nulo	Tipo de Atributo
Lugar	Número do Lugar do Cliente	<i>String</i> variável até 4 carateres	Não	Identificador (Simples)
Sala	Número de identificação da Sala	Número inteiro positivo até 2 algarismos	Não	Simples
Sessão	Hora de início do Filme	TIME	Não	Simples

Tabela 5 :Caracterização dos Atributos da Entidade InfoBilhete

Entidade: Filme				
Atributo	Descrição	Tipo de Dados e Domínio	Nulo	Tipo de Atributo
IdFilme	Número identificador do Filme que corresponde ao IDMB iD	Número inteiro positivo até 7 algarismos	Não	Identificador (Simples)
Nome	Nome do Filme	String variável até 45 caracteres	Não	Simples
Ano	Ano de estreia do Filme	YEAR	Não	Simples
Duração	Duração total do Filme	TIME	Não	Simples
País	País em que o Filme foi produzido	String variável até 2 caracteres (ISO_ALPHA2)	Sim	Simples
Realizador	Realizador do Filme	String variável até 45 caracteres	Sim	Simples
Elenco	Conjunto dos Nomes dos atores que participaram no Filme	TEXT	Sim	Simples
Sinopse	Breve resumo do Filme	TEXT	Sim	Simples
Género	Género em que o Filme se enCaixa	String variável até 45 carateres	Não	Multivalor

Tabela 6 :Caracterização dos Atributos da Entidade Bilhete

3.5. Detalhe ou generalização de Entidades

O modelo conceptual inicial sofreu algumas generalizações, mas também detalhes de certas informações base pedidas pelo nosso empregador.

- **1ª Modificação:**

A primeira modificação realizada foi no sentido de criar uma Entidade para aglomerar a Informação do Bilhete que anteriormente estavam definidas como Atributos únicos do mesmo. Como o Lugar, Sessão e Sala são elementos que necessitam de se verificar corretos e únicos apenas em grupos de ternos, apostou-se no detalhe para outra Entidade, de forma a futuramente se poder verificar e confirmar que um Bilhete não pode ter, em simultâneo, uma Sala, Sessão e Lugar iguais.

- **2ª Modificação:**

A segunda modificação realizada foi a de eliminar o Atributo NIF que pertencia à Entidade Bilhete e criar uma nova Entidade Cliente que para além de possuir este Atributo (passa a ser o identificador) ganha também outros Atributos como Data de Nascimento e Contacto, tal como pedido pelo empregador.

3.6. Apresentação e explicação do Diagrama ER

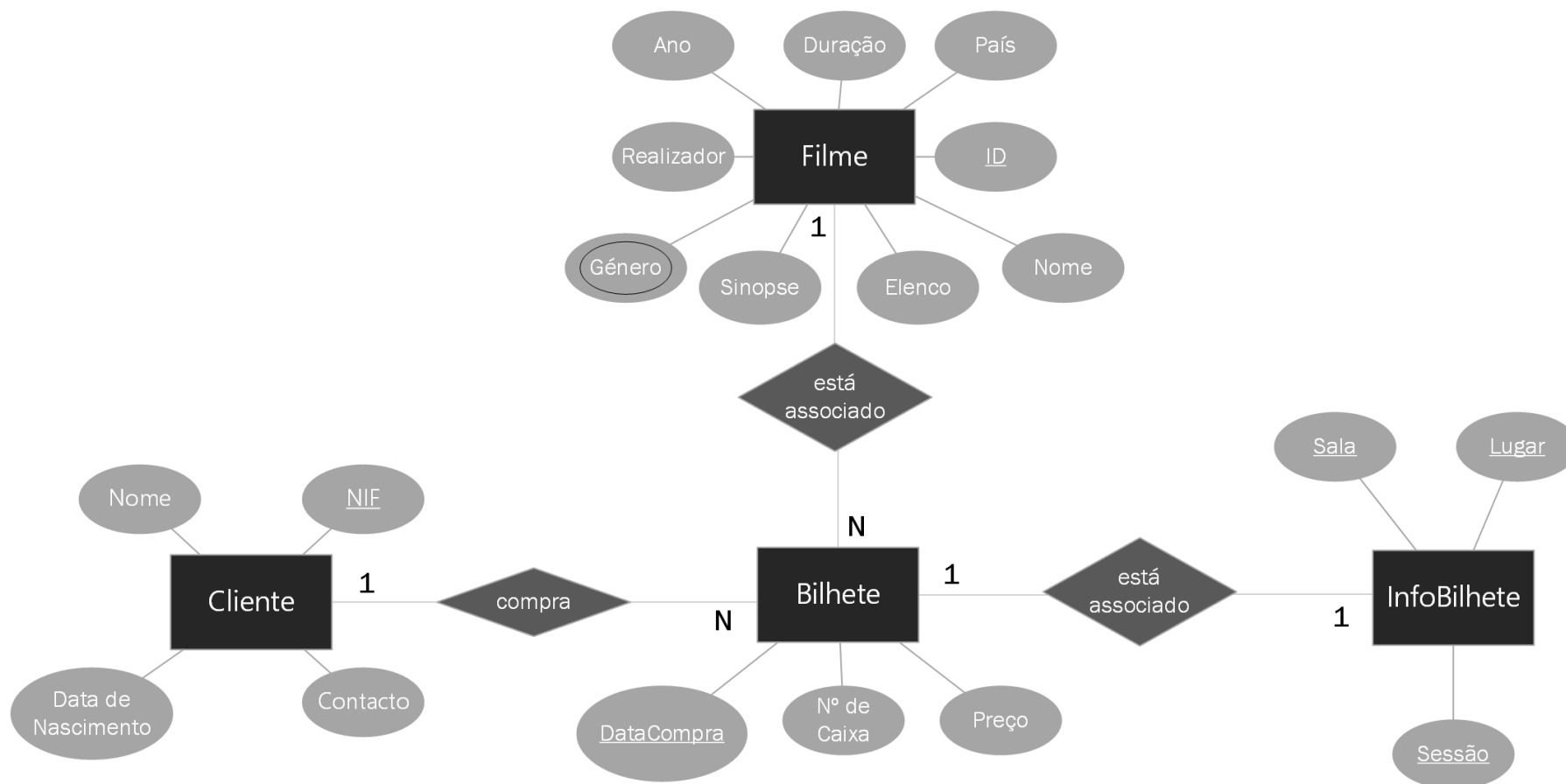


Figura 1: Diagrama ER (Modelo Concetual)

O modelo conceptual foi começando a ganhar vida através das ideias base recolhidas desde o começo da realização do nosso modelo. Focamos para isso, de imediato, nas Entidades que eram indispensáveis existir e consequentemente na projeção dos respetivos Atributos.

Sendo a nossa BD alusiva a uma Bilheteira de Cinema definimos as seguintes Entidades no nosso modelo concetual:

- A primeira Entidade pensada foi o Bilhete, que retrata no fundo a Entidade principal da nossa BD, uma vez que, é através dela que toda a ideia do funcionamento da Bilheteira é contida. É ainda por meio desta Entidade que o Cliente é informado não só do seu Lugar na respetiva Sessão e Sala, mas também do seu preço e da sua Data de Compra.

Em adjacência ao Bilhete (relação 1:1) foi criada uma Entidade que aglomera a informação do Bilhete que é dependente das próprias tais como Sala, Lugar e Sessão. Dessa forma, podemos assegurar que esses dados são únicos de terno em terno.

- A segunda Entidade adveio da Entidade supracitada e diz respeito ao Filme. Esta foi criada para obtermos os dados referentes aos Filmes em cartaz ou passados no nosso Cinema de forma a possibilitar uma gestão dessa informação para o ganho interno da empresa.
- A terceira Entidade serviu para informatizar os dados dos diversos Clientes que efetuaram uma ou mais compras na Bilheteira para não só ser mais fácil a reutilização desses dados para uso futuro, mas também para a análise e/ou marketing do Cinema.

3.7. Validação do modelo de dados com o utilizador

Numa segunda entrevista realizada com o empregador, após toda a contextualização, criação e modelação conceptual, fez-se de uma forma bastante intuitiva a verificação geral da relação entre as nossas Entidades/Atributos e os requisitos pedidos pelo empregador. Manteve-se a ideia duma BD simples, funcional e que respondesse tanto à escalabilidade da Bilheteira como também à simplicidade da infraestrutura do Cinema em causa. Todas as explorações pedidas pelo Cliente também se verificam a nível conceitual (em que mais tarde serão aprofundados em linguagem SQL). Dessa forma, o modelo encontra-se verificado e aprovado pelo empregador.

4. Modelação Conceptual

4.1. Construção e validação do modelo de dados lógico

Depois de elaborado o modelo conceptual avançamos, assim, para a realização do modelo lógico.

Partindo da análise conceptual, começamos por avaliar a necessidade da criação de diferentes tabelas para guardar os dados requeridos pelo nosso empregador. Como o nosso Cliente precisava de armazenar informações relativas aos diferentes Filmes exibidos no Cinema foi criado uma Tabela Filme. Nesta, o Atributo da chave primária é o IDFilme que corresponde ao *IMDBiD*, o que nos permite que cada Filme tenha uma entrada na nossa BD única. Usamos o tipo de dados *MEDIUMINT(7)*, por forma a truncar o valor para os 7 algarismos usados pelo IMDB e assim poupar espaço em disco futuramente. Para cada Filme podemos armazenar ou não o Nome do Realizador, um pequeno texto descritivo sobre o Elenco, a Sinopse e também usar o ISO_ALPHA2 para identificar um País com 2 caracteres apenas. Para além disto armazenamos, obrigatoriamente, o Nome do Filme, a sua Duração e o Ano de Estreia. Sendo o Género um Atributo multivalor na modelação conceptual e partindo da necessidade de armazenar *n* Géneros que depois catalogarão os Filmes em si, criou-se a Tabela Género com uma chave primária IDGénero. Esta é necessária para garantir a singularidade de cada Género introduzido no Sistema. Como Atributo simples temos, logicamente, o Nome do Género. Com a introdução da chave estrangeira que fará a ligação para a tabela do Filme, teremos assim relacionado o Género para cada Filme.

Posteriormente, o empregador necessitava de armazenar os dados referentes aos Clientes do Cinema. Deste modo, criamos uma tabela Cliente em que começamos por identificar o Atributo candidato à chave primária que corresponde, naturalmente, ao NIF pois, deste modo, garantimos que não existem diferentes entradas para o mesmo Cliente. Posteriormente, definimos como Atributos simples o Nome do Cliente, o Contacto e a Data de Nascimento. Utilizamos o tipo de dados *INT(9)* de forma a truncar para os 9 algarismos significativos tanto no NIF como no nº de telemóvel.

De seguida, surgiu a necessidade de informações relativas aos Bilhetes que estes adquiriram. Assim, surge uma nova Tabela Bilhete que se relaciona com a Tabela Cliente. Esta relação visa a que para cada Cliente seja armazenada informações relativas aos diferentes Bilhetes adquiridos pelo mesmo. Em cada Bilhete pretendeu-se guardar informações como a Sala, o Filme, o número de Lugar, a Sessão, o preço e, opcionalmente, o número de Caixa onde foi vendido. Neste caso, a chave primária é composta e surge das relações existentes. A chave primária DataCompra corresponde, obviamente, à Data de Compra do Bilhete que vai até ao nível temporal do segundo, assegurando, assim, a sua

unicidade. Para além deste, temos como chave primária e estrangeira o IDFilme que assegura a correspondência obrigatória de um Bilhete a um Filme. Temos também como chave primária e estrangeira o NIF do Cliente que permite corresponder a venda de um determinado Bilhete a um Cliente. Por fim, temos como chaves estrangeiras e primárias a Sala, a Sessão e o Lugar que são herdadas da relação unária para a tabela InfoBilhete. A tabela InfoBilhete surge no sentido em que os Atributos Sessão, Lugar e Sala eram dotados de uma dependência pois um Lugar, de uma determinada Sessão, de uma determinada Sala só pode ser vendido a um Cliente numa dada data. De forma a garantir o que foi anteriormente supracitado, criou-se uma tabela que possui a chave primária composta constituída pela Sala (nº de Sala), Sessão (hora da Sessão) e Lugar (nº de Lugar), dessa forma garantindo que os Bilhetes possam ter Salas, Sessões e Lugares diferentes, mas também Lugares iguais para sessões/Salas diferentes. Apenas se torna errado quando vários Bilhetes para o mesmo dia e Filme tenham a mesma Sala, Sessão e Lugar, sendo lógico essa recusa.

4.2. Desenho do Modelo Lógico

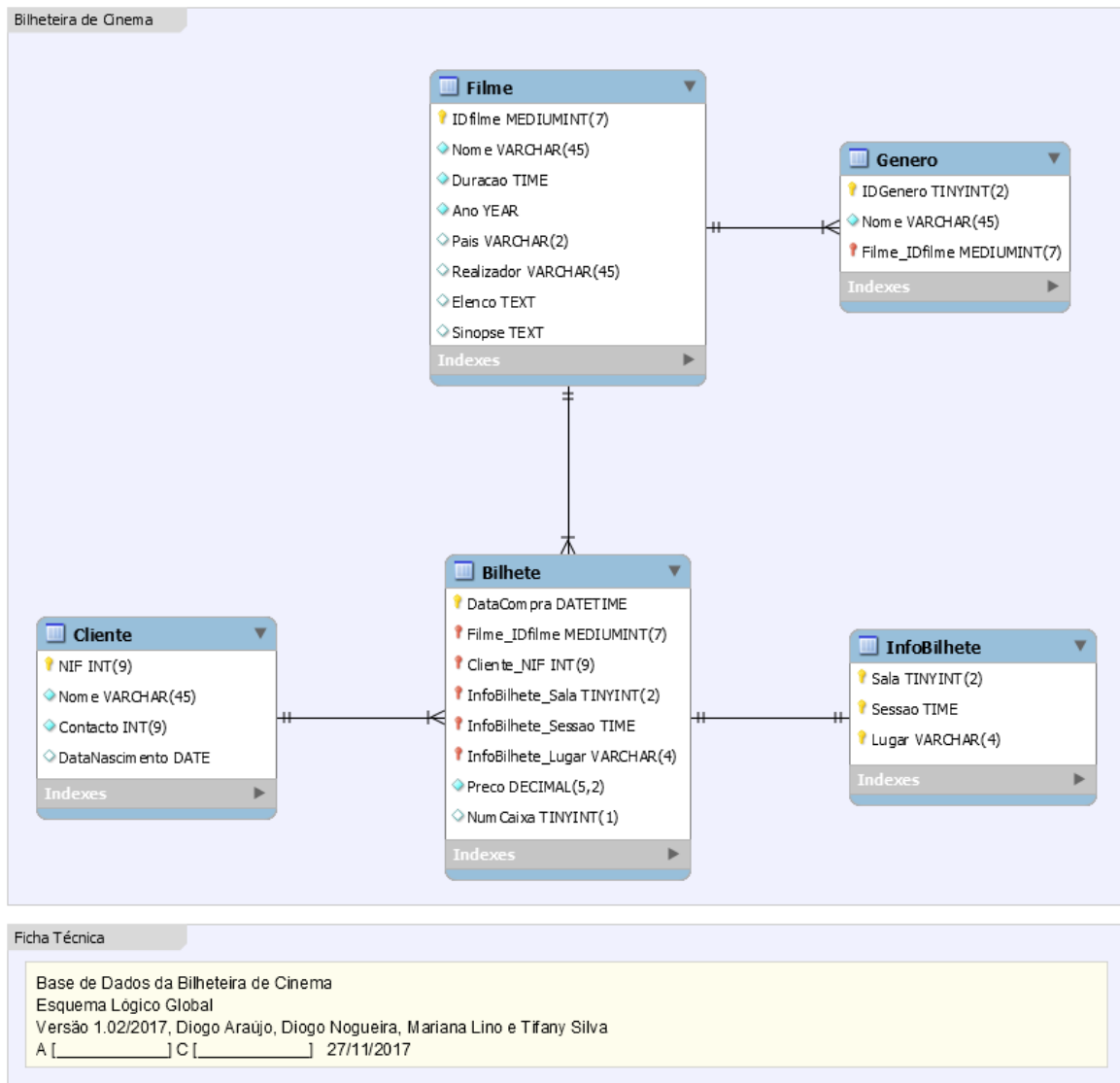


Figura 2: Modelo Lógico

4.3. Validação do modelo através da normalização

Com as várias formas de normalização torna-se mais fácil e intuitivo criar uma BD que seja consistente e funcional, acabando por a tornar mais simples e mais perceptível. No projeto que desenvolvemos utilizamos essas três formas de normalização que, de seguida, enumeraremos. Vamos, para isso, usar como suporte exemplos onde acabamos por aplicar as mesmas.

- **1FN – 1ª Forma Normal:**

A primeira forma normal define-se quando uma tabela não contém nenhum grupo de dados repetidos, isto é, quando todos os valores são únicos.

A nossa BD tem em parte a 1ª Normalização feita. No caso da tabela Cliente ou mesmo Género todos os Atributos têm valores únicos, os quais apenas se relacionam com o Cliente e com o Género, respetivamente. A tabela Filme e Bilhete terão dados repetidos dado que por exemplo existem vários Filmes realizados no mesmo Ano, País ou até pelo mesmo Realizador. Em analogia, o Bilhete também pode ter o mesmo NIF caso o Cliente compre mais que um, vários Bilhetes ter o mesmo preço ou até terem sido comprados na mesma Caixa.

- **2FN – 2ª Forma Normal:**

Uma tabela diz-se estar na segunda forma normal, se estiver na 1FN e se todos os Atributos não-chave forem totalmente dependentes da chave primária.

Na nossa BD, apesar de algumas tabelas não terem a 1FN temos na tabela Cliente, Filme, Género todos os Atributos associados dependentes/derivados das chaves primárias respetivas. Como exemplos todos os Atributos do Filme estão dependentes do ID que fornecemos ao mesmo, e o Nome e Contacto do Cliente estão associados ao NIF.

- **3FN – 3ª Forma Normal:**

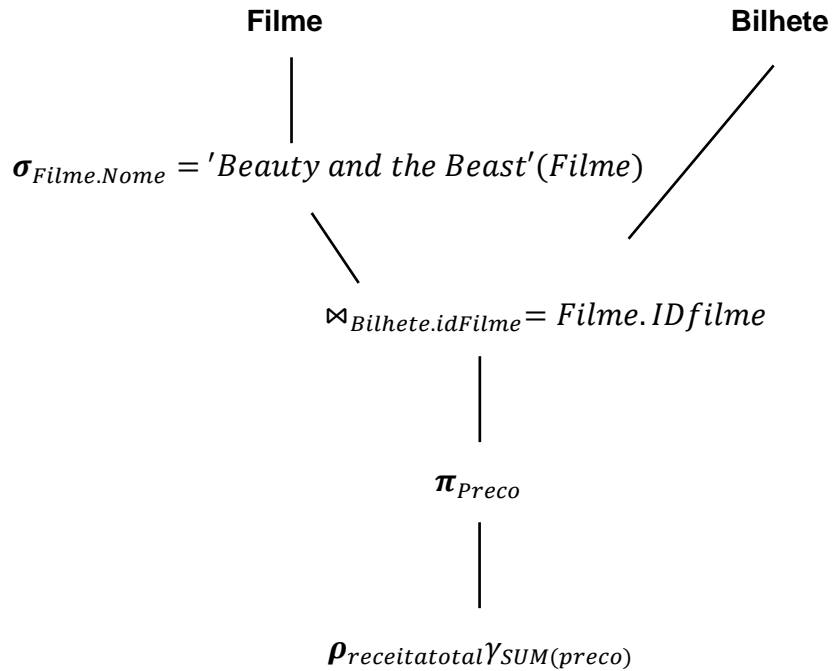
Uma tabela está na terceira forma normal, quando estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.

Na nossa BD surgiu ao construir o modelo lógico uma dependência de dois Atributos da classe Cliente que eram o Lugar e Sessão. Dado que eram Atributos não chave e ambos se dependiam um do outro (dado que um certo Lugar está ou não ocupado para uma determinada Sessão) decidimos, portanto, colocar todos esses Atributos dependentes numa tabela separada ligada com uma ligação de 1:1 chamada InfoBilhete. Esta continha os Atributos Sala, Sessão e Lugar em que todos formavam a chave primária composta. Por conseguinte, esses Atributos já dependiam apenas de Atributos chave e vice-versa.

4.4. Validação do modelo com as interrogações do utilizador

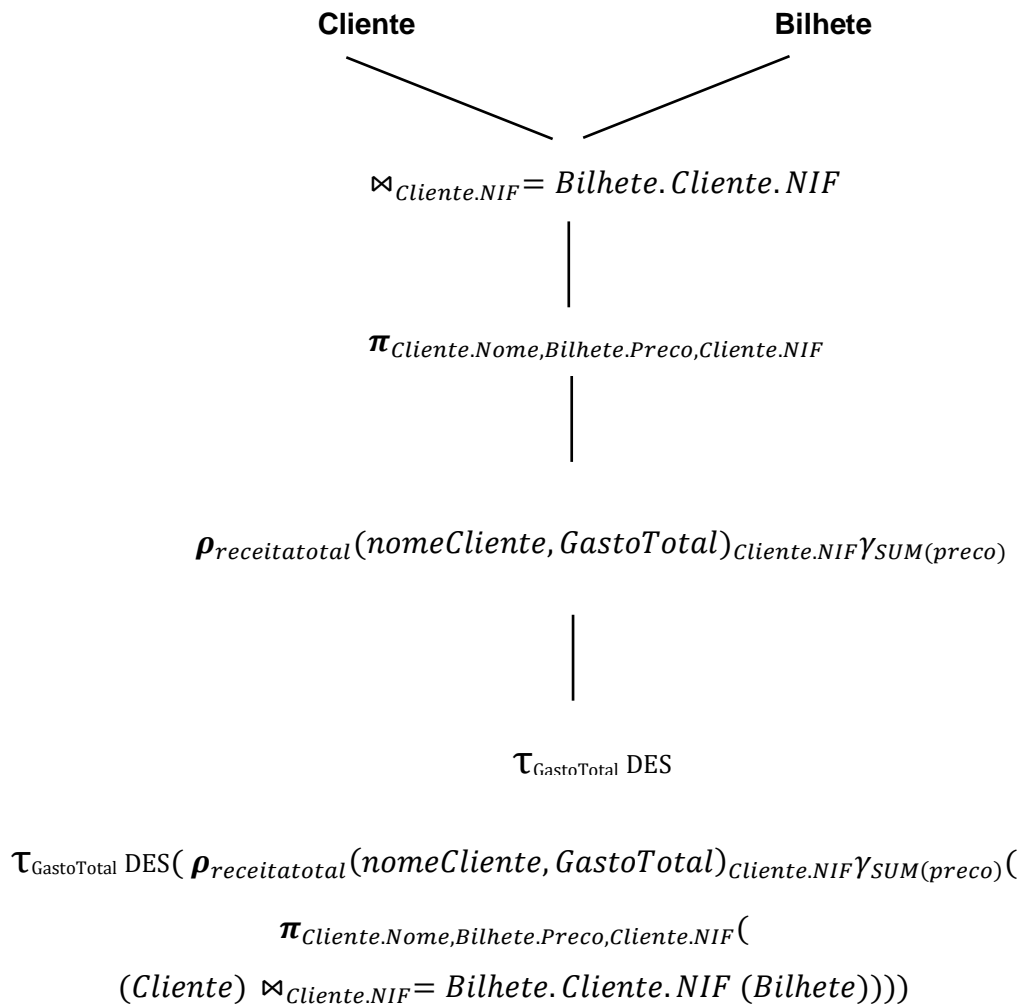
Para procedermos à validação do nosso modelo usando como apoio as interrogações desde início efetuadas pelo utilizador/empregador, vamos retorquir duas das mesmas, de forma específica e com base no modelo projetado.

- Saber a receita total de um certo Filme como, por exemplo, *Beauty and the Beast*:



$$\rho_{Preceittotal} \gamma_{SUM}(preco) (\pi_{Preco} (\sigma_{Filme.Nome = 'Beauty and the Beast'}(Filme) \bowtie_{Filme.IDfilme = Bilhete.IDfilme} Bilhete))$$

- Listar os Nomes dos Clientes e respectivos gastos totais de compras de Bilhetes, por ordem decrescente dos preços totais:



4.5. Validação do modelo com as transações estabelecidas

De forma a melhorar a performance da BD, foi necessária uma análise das transações que ocorrem entre as relações. A partir desta análise, foi possível perceber a sua funcionalidade e identificar as transações mais importantes. Estabelecemos algumas *queries* que para além da necessidade de usar seleções, tem a precisão também de usar a inserção e atualização.

Optamos então por criar um mapa de transações de forma a identificar o tipo de operações que cada *query* implica. A partir da análise do modelo lógico, detetamos algumas transações possíveis:

- **Inserir os detalhes de um novo Filme:**

Além de se querer inserir os dados relativos a novo Filme na base dados é necessário também atualizar a Tabela Genero caso o(s) Género(s) inseridos não existam ou que já existam e precisam de ser associados. Assim, avalia-se primeiro e vamos pressupor que são vários Géneros existem na tabela. Se existir, cria-se uma nova entrada associando o Nome do Género ao id do Filme. Caso não exista, cria-se uma nova tabela fornecendo o novo Nome do Género e associando o id do Filme.

Se não ocorrer nenhum erro, a transação é realizada e o *commit* é executado com sucesso. Caso contrário, as modificações realizadas na base de dados são desfeitas, voltando esta ao seu estado inicial.

- **Atualizar o Contacto de um Cliente :**

Para além de atualizar um Contacto de um determinado Cliente, pressupomos que este irá comprar posteriormente um Bilhete para um determinado Filme. Antes de inserir na tabela Bilhete uma nova entrada temos que assegurar se a combinação Sala, Lugar, Sessão existe. Caso exista, não é necessário criar nenhuma entrada para tabela InfoBilhete. Caso contrário, é necessário fazer a inserção desta nova combinação (Sala, Lugar, Sessão) na tabela InfoBilhete.

Assim, depois de elaborado o que foi dito anteriormente podemos fazer a inserção do novo Bilhete na tabela Bilhete.

Caso não haja nenhum erro é feito o *commit*, caso contrário são desfeitas todas as alterações realizadas na base de dados.

4.6. Revisão do modelo lógico com o utilizador

Nesta fase de desenvolvimento do modelo lógico definimos como principal objetivo a revisão deste mesmo, contudo, desta vez, de acordo com a visão do utilizador/empregador de modo a garantir que o nosso modelo corresponde ao que desde início foi instituído cumprir.

Após a revisão do nosso modelo lógico com o utilizador, conseguimos dá-lo como aceite uma vez que, cumpre os demais requisitos inicialmente previstos e estabelecidos.

5. Implementação Física

5.1. Seleção do Sistema de gestão de BD

O principal motivo para a escolha do Sistema MySQL para a realização da nossa BD passou pelo facto de esta ser a BD *open-source* número um sendo, por essa razão, uma excelente escolha para embarcar no nosso projeto dado que assim não obrigamos o nosso empregador a despende de servidores e licenças pagas (*closed-source*) para outras escolhas de RDBMS (MySQL, 2017).

Aliando isto à sua consistência, alta disponibilidade, performance e também confiabilidade, torna-se um Sistema bastante fácil e intuitivo usar. Além disso, funciona em mais de 20 plataformas e é utilizado por grandes empresas de *software* como a *Google*, *Facebook* e *YouTube*.

5.2. Tradução do esquema lógico para o Sistema de gestão de bases de dados escolhido em SQL

```
-- Criação em SQL da BD
-- Diogo Araújo, Diogo Nogueira, Mariana Lino e Tifany Silva

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-----
-- Libertar um SCHEMA com o Nome CINEMA já existente
-----

DROP SCHEMA IF EXISTS `Cinema` ;

-----
-- Criar Schema Cinema
-----

CREATE SCHEMA IF NOT EXISTS `Cinema` DEFAULT CHARACTER SET utf8 ;
USE `Cinema` ;

-----
-- Tabela `Cinema`.`Filme`
-----

CREATE TABLE IF NOT EXISTS `Cinema`.`Filme` (
  `IDFilme` MEDIUMINT(7) UNSIGNED NOT NULL COMMENT 'Id IMDB ',
  `Nome` VARCHAR(45) NOT NULL,
  `Duracao` TIME NOT NULL,
  `Ano` YEAR NOT NULL,
  `Pais` VARCHAR(2) NULL COMMENT 'ISO_ALPHA2',
  `Realizador` VARCHAR(45) NULL,
  `Elenco` TEXT NULL,
  `Sinopse` TEXT NULL,
  PRIMARY KEY (`IDFilme`))
ENGINE = InnoDB;

-----
-- Tabela `Cinema`.`Cliente`
-----

CREATE TABLE IF NOT EXISTS `Cinema`.`Cliente` (
  `NIF` INT(9) UNSIGNED NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Contacto` INT(9) UNSIGNED NOT NULL,
  `DataNascimento` DATE NULL,
  PRIMARY KEY (`NIF`))
ENGINE = InnoDB;

-----
-- Tabela `Cinema`.`InfoBilhete`
```

```

-----
CREATE TABLE IF NOT EXISTS `Cinema`.`InfoBilhete` (
  `Sala` TINYINT(2) UNSIGNED NOT NULL,
  `Sessao` TIME NOT NULL,
  `Lugar` VARCHAR(4) NOT NULL,
  PRIMARY KEY (`Sala`, `Sessao`, `Lugar`))
ENGINE = InnoDB;
-----
-- Tabela `Cinema`.`Bilhete`
-----
CREATE TABLE IF NOT EXISTS `Cinema`.`Bilhete` (
  `DataCompra` DATETIME NOT NULL,
  `Filme_IDFilme` MEDIUMINT(7) UNSIGNED NOT NULL,
  `Cliente_NIF` INT(9) UNSIGNED NOT NULL,
  `InfoBilhete_Sala` TINYINT(2) UNSIGNED NOT NULL,
  `InfoBilhete_Sessao` TIME NOT NULL,
  `InfoBilhete_Lugar` VARCHAR(4) NOT NULL,
  `Preco` DECIMAL(5,2) NOT NULL,
  `NumCaixa` TINYINT(1) UNSIGNED NULL,
  PRIMARY KEY (`DataCompra`, `Filme_IDFilme`, `Cliente_NIF`,
`InfoBilhete_Sala`, `InfoBilhete_Sessao`, `InfoBilhete_Lugar`),
  INDEX `fk_Bilhete_Filme1_idx` (`Filme_IDFilme` ASC),
  INDEX `fk_Bilhete_Cliente1_idx` (`Cliente_NIF` ASC),
  INDEX `fk_Bilhete_InfoBilhete1_idx` (`InfoBilhete_Sala` ASC,
`InfoBilhete_Sessao` ASC, `InfoBilhete_Lugar` ASC),
  CONSTRAINT `fk_Bilhete_Filme1`
    FOREIGN KEY (`Filme_IDFilme`)
    REFERENCES `Cinema`.`Filme` (`IDFilme`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Bilhete_Cliente1`
    FOREIGN KEY (`Cliente_NIF`)
    REFERENCES `Cinema`.`Cliente` (`NIF`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Bilhete_InfoBilhete1`
    FOREIGN KEY (`InfoBilhete_Sala`, `InfoBilhete_Sessao`,
`InfoBilhete_Lugar`)
    REFERENCES `Cinema`.`InfoBilhete` (`Sala`, `Sessao`, `Lugar`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
-----
-- Tabela `Cinema`.`Genero`
-----
CREATE TABLE IF NOT EXISTS `Cinema`.`Genero` (
  `IDGenero` TINYINT(2) UNSIGNED NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Filme_IDFilme` MEDIUMINT(7) UNSIGNED NOT NULL,

```



```

PRIMARY KEY (`IDGenero`, `Filme_IDFilme`),
INDEX `fk_Genero_Filme_idx` (`Filme_IDFilme` ASC),
CONSTRAINT `fk_Genero_Filme`
  FOREIGN KEY (`Filme_IDFilme`)
  REFERENCES `Cinema`.`Filme` (`IDFilme`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

5.3. Tradução das interrogações do utilizador para SQL

- Saber a receita total de um certo Filme como, por exemplo, *Beauty and the Beast*:

```

-- Saber a receita total do Filme Beauty and the Beast
SELECT SUM(Preco) FROM Bilhete
JOIN Filme ON Bilhete.Filme_IDFilme = Filme.IDFilme
WHERE (Filme.Nome = "Beauty and the Beast");

```

- Listar os Nomes dos Clientes e respetivos gastos totais nas compras do Bilhete, por ordem decrescente de preços totais:

```

-- Saber o top 3 de Clientes que mais gastaram no Cinema
SELECT Nome, SUM(Preco) AS GastoTotal FROM Cliente
JOIN Bilhete ON Cliente.NIF = Bilhete.Cliente_NIF
GROUP BY NIF
ORDER BY GastoTotal DESC;

```

5.4. Tradução das transações estabelecidas para SQL(alguns exemplos)

- Atualizar o Contacto de um Cliente :

```
DELIMITER //
```

```
CREATE PROCEDURE updateContactoNovoBilhete(IN nif INT, IN contactonovo  
INT, IN idfilme mediumint(7),IN sala tinyint(2),  
IN sessao time, IN lugar varchar(4), IN caixa tinyint(1),IN preco  
double(5,2))
```

```
BEGIN  
  DECLARE erro BOOLEAN DEFAULT 0;  
  DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET Erro =1;  
  START TRANSACTION;  
  -- Verificar o estado inicial do registo  
  SELECT Contacto FROM cliente  
    WHERE NIF = nif;  
  -- Alterar o contacto do cliente com aquele nif  
  UPDATE cliente  
    SET Contacto = contactonovo  
    WHERE NIF = nif;  
  -- Inserção de um nova entrada na tabela infobilhete caso não exista  
  esta entrada  
  IF ( (SELECT COUNT(*) FROM infobilhete WHERE Sala = sala AND  
Sessao=sessao AND Lugar=lugar) = 0) THEN  
    INSERT INTO infobilhete VALUES(sala,sessao,lugar);  
  END IF;  
  -- Inserção de uma nova entrada na tabela bilhete  
  INSERT INTO bilhete VALUES("2017-09-10  
17:15:30",nif,idfilme,sala,sessao,lugar,preco);  
  
  IF Erro THEN ROLLBACK;  
  END IF;  
  COMMIT;  
END //
```

5.5. Escolha, definição e caracterização de índices em SQL

Um índice é definido como uma estrutura de dados que permite ao SGBD localizar registos num ficheiro de forma mais rápida e eficaz e, com isso, melhorar o tempo de resposta às interrogações definidas pelo utilizador.

Na nossa BD, os únicos índices que existem são as próprias chaves primárias/estrangeiras associadas às tabelas por nós projetadas. Assim sendo, acabamos por não ter a necessidade de definir índices extras de modo a melhorar o tempo das nossas queries pelo principal motivo de não ser necessário perante uma BD relativamente pequena como a nossa.

5.6. Estimativa do espaço em disco da BD e taxa de crescimento anual

O motor de armazenamento utilizado para esta BD é o *InnoDB* do *MySQL*. Assim sendo, inicialmente, procuramos conhecer o espaço que ocupa cada tipo de domínio utilizado na BD.

A partir de uma pesquisa no site oficial da linguagem *MySQL* (MySQL, 2017), apuramos as seguintes informações:

- Integer 4 bytes
- Tinyint(2) 1 bytes
- Mediumint 3 bytes
- Decimal(5,2) 4 bytes
- Date 3 bytes
- Datetime 5 bytes
- Time 3 bytes
- Year 1 byte
- Varchar(N) 1+N bytes
- Text 2¹⁶+2 bytes

Partindo da seguinte *query* podemos conhecer o espaço que cada tabela da nossa BD ocupa (em KB):

```
SELECT table_name ,
       round(((data_length + index_length) )/1024, 2) as SIZE_KB
FROM information_schema.TABLES
WHERE table_schema = (SELECT DATABASE()) ORDER BY SIZE_KB DESC;
```

	table_name	SIZE_KB
1	bilhete	64.00
2	genero	32.00
3	filme	16.00
4	cliente	16.00
5	infobilhete	16.00

Tabela 7 :Tamanho que cada tabela ocupa em KB

Partindo da tabela acima citada, calculamos que a BD, no total, e sem qualquer tipo de dados inseridos, ocupa 144 KB.

Assim, podemos fazer uma estimativa inicial consoante o nº de entradas de cada tabela, para um valor inicial da BD povoada:

- Se tivermos 1000 entradas para a tabela Bilhete, o tamanho total da tabela seria:
 $1000 \times 64 = 64000$ KB;
- Se tivermos 30 entradas para a tabela Genero, o tamanho total da tabela seria:
 $30 \times 32 = 960$ KB;
- Se tivermos 15 entradas para a tabela Filme, o tamanho total da tabela seria:
 $15 \times 16 = 240$ KB;
- Se tivermos 1000 entradas para a tabela Cliente, o tamanho total da tabela seria:
 $1000 \times 16 = 16000$ KB;
- Se tivermos 1100 entradas para o InfoBilhete, o tamanho total da tabela seria:
 $1100 \times 16 = 17600$ KB.

Fazendo os devidos cálculos, no total, a BD ocuparia 98800 KB.

Ao fim de 1 Ano e supondo que a taxa de crescimento seria de 25%, obtemos uma estimativa do espaço em disco da BD que corresponde a:

$$98800 + 0.25 \times 98800 = 123500 \text{ KB } (\sim 121 \text{ MB})$$

5.7. Definição e caracterização dos mecanismos de segurança em SQL

Foram definidos e implementados mecanismos de segurança em SQL de modo a fornecer permissões de acesso/alteração da BD acabando por garantir a proteção da mesma. Apresentamos assim de seguida os mecanismos criados:

- **Criação de 2 utilizadores:** um ‘admin’ que corresponde ao administrador do Cinema e o “caixeiro” que corresponde ao funcionário que vende os Bilhetes:

```
-- Criação de os utilizadores e as suas respetivas permissões
CREATE USER 'CinemaAdmin'@'localhost' IDENTIFIED BY 'admin';
CREATE USER 'Caixeiro'@'localhost' IDENTIFIED BY 'caixeiro';
```

- **Atribuição de permissões ao “caixeiro” inserção, eliminação ou *update* de informações relativas ao Cliente, Bilhete e InfoBilhete:**

```
GRANT ALL PRIVILEGES ON Cinema TO 'CinemaAdmin'@'localhost';
GRANT INSERT, UPDATE, DELETE ON Cliente to 'Caixeiro'@'localhost';
GRANT INSERT, UPDATE, DELETE ON Bilhete to 'Caixeiro'@'localhost';
GRANT INSERT, UPDATE, DELETE ON InfoBilhete to
'Caixeiro'@'localhost';
FLUSH PRIVILEGES;
```

- **Atribuição de permissões ao “admin” de eliminação de Clientes:**

```
-- Eliminar os utilizadores criados
DROP USER 'CinemaAdmin'@'localhost';
```

5.8. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

Decidimos definir uma *view* para uma das interrogações feitas anteriormente pelo nosso utilizador. Para efeito, apresentamos de seguida o código em SQL relativo à *view* em questão bem como a interrogação à qual corresponde.

- **Tabela que apresenta por ordem decrescente os gastos totais de todos os Cliente e ainda o respetivo Nome de cada um:**

```
CREATE VIEW GastoTotal AS
  SELECT Nome, SUM(Preco) AS GastoTotal FROM Cliente
    JOIN Bilhete ON Cliente.NIF= Bilhete.Cliente_NIF
    GROUP BY NIF
    ORDER BY GastoTotal DESC;
SELECT * FROM GastoTotal;
```

5.9. Revisão do Sistema implementado pelo utilizador

Neste último capítulo foi abordada a modelação física e com ela, fomos capazes de povoar a nossa BD, bem como aplicar, em SQL, parte dos requisitos que antes foram definidos essenciais para a BD em questão. Ou seja, fizemos a tradução do esquema lógico para o Sistema de gestão de BD e, posteriormente abordamos outros tópicos que precisavam ser revistos para assegurar que todo o Sistema fosse dado como aprovado.

Após toda uma revisão do Sistema com o utilizador, e perante tudo o que foi feito para tal, conseguimos dá-lo como aceite uma vez que, cumpre os demais requisitos inicialmente previstos e estabelecidos.

6. Conclusões e trabalho futuro

Para se proceder ao correto e funcional desenvolvimento de uma BD, tem de se percorrer um conjunto de passos fundamentais. Este desenvolvimento é um processo sequencial que exige que seja definido desde o começo um conjunto de informações base e detalhadas de acordo com o pretendido. Iniciamos assim com uma contextualização do projeto em causa, avançamos para interpretação específica e acabamos na construção e organização dos 3 tipos de modelos necessários à construção desta.

Desta forma, com a definição do Sistema inicial fomos capazes de particularizar o problema fundamental a ser tratado usando como suporte uma introdução daquilo que teríamos em mão, numa fase posterior, definindo e analisando tudo o que o nosso utilizador citou como necessário. Para que todos os requisitos fossem respondidos, criamos as Entidades necessárias, os seus respetivos Atributos e estabelecemos os relacionamentos mais corretos entre as mesmas de uma forma credível e esquematizada. No decorrer do crescimento do projeto, toda a fase de criação de modelos passou por algumas transformações de acordo com as dificuldades que foram surgindo. Fomos alterando o modelo conceitual inicial, e consequentemente o modelo lógico e físico, de modo a garantirmos que todos estes respondessem aos requisitos pretendidos e ao mesmo tempo fossem concretos de compreender.

Achamos importante ainda referir que, a nossa BD é facilmente extensível para outros tipos de Bilheteiras (como por exemplo, uma Bilheteira de qualquer outro tipo de evento) que o utilizador possa eventualmente querer adotar, já que, é uma BD bastante simples, mas ao mesmo tempo capaz de cumprir todos os requisitos essenciais de uma Bilheteira.

7. Referências Bibliográficas

- Data Type Storage Requirements.* (2017). Obtido de MySQL:
<https://dev.mysql.com/doc/refman/5.7/en/storage-requirements.html#data-types-storage-reqs-numeric>
- Cinematography.* (07 de Novembro de 2017). Obtido de Wikipedia:
<https://en.wikipedia.org/wiki/Cinematography>
- MySQL.* (22 de Novembro de 2017). Obtido de Wikipedia: <https://en.wikipedia.org/wiki/MySQL>

8. Lista de Siglas e Acrónimos

BD: Base de Dados

RDBMS: Relational Database Management System

ER: Entidade-Relacionamento

SQL: Structured Query Language

SGBD: Sistema de Gestão de Base de Dados