



**Universidade do Minho**  
Licenciatura em Ciências da Computação

## **Unidade Curricular de Bases de Dados**

Ano Lectivo de 2021/2022

### **Deixa Ver**

**João Marques – a84684**

**Rui Baptista – a87989**

**Rui Sousa – a85908**

**Miguel Gonçalves – a90416**

Dezembro, 2021

# **BD**

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Deixa Ver

**João Marques – a84684**

**Rui Baptista – a87989**

**Rui Sousa – a85908**

**Miguel Gonçalves – a90416**

Dezembro, 2021

## Resumo

Este trabalho prático foi proposto no âmbito da unidade curricular Base de Dados, tendo como por objetivo a análise, planeamento, modelação, arquitetura e implementação de um Sistema de Base de Dados, com o tema “Deixa Ver”.

O presente relatório visa detalhar toda a elaboração da base de dados em questão além de detalhar todas as etapas da sua criação e o seu modelo lógico criado a partir de um levantamento bastante sucinto de dados relativos ao tema em questão.

**Área de Aplicação:** Modelação e implementação de Sistemas de Bases de Dados.

**Palavras-Chave:** Base de dados, Modelo Conceptual

# Índice

1. Introdução	1
1.1 Contextualização	1
1.2 Apresentação do Caso de Estudo	1
1.3 Motivação e Objetivos	2
1.4 Estrutura do Relatório	2
2. Levantamento e Análise de Requisitos	3
2.1 Método de levantamento e de análise de requisitos	3
2.2 Requisitos Levantados	3
2.2.1 Requisitos de descrição	3
2.2.2 Requisitos de exploração	4
2.2.3 Requisitos de controlos	4
2.3 Análise e validação geral dos requisitos	5
3. Modelo Conceptual	5
3.1 Apresentação da abordagem de modelização realizada	5
3.2 Identificação e caracterização das entidades	6
3.3 Identificação e caracterização dos relacionamentos	7
3.4 Identificação e caracterização da associação dos Atributos com as Entidades e Relacionamentos	8
3.5 Detalhe ou generalização de entidades	15
3.6 Diagrama ER	16
4. Modelação Lógica	17
4.1 Construção e validação do modelo de dados lógico	17
4.2 Desenho do modelo lógico	19
4.3 Validação do modelo através da normalização	20
4.4 Validação do modelo com interrogações do utilizador	21
4.5 Conjunto de procedimentos em MySQL	24
4.6 Escolha, definição e caracterização de índices em SQL	27

5. MongoDB	28
5.1 Migração dos dados do sistema relacional para o sistema de dados em MongoDB	28
5.2 Conjunto de queries em MongoDB	29
6. Análise crítica do trabalho realizado e Conclusão	32
7. Bibliografia	34

## Índice de Figuras

Figura 1 – Diagrama ER	16
Figura 2 – Modelo Lógico	19
Figura 3 – Árvore histórico do cliente através do NIF	21
Figura 4 – Árvore obter produção através do título	22
Figura 5 – Árvore compras numa dada data	22
Figura 6 – Árvore Top N séries/filmes	23
Figura 7 – Código SQL Correspondente alínea a)	24
Figura 8 – Código SQL Correspondente alínea b)	24
Figura 9 – Código SQL Correspondente alínea c)	25
Figura 10 – Código SQL Correspondente alínea d)	25
Figura 11 – Código SQL Correspondente alínea e)	25
Figura 12 – Código SQL Correspondente alínea f)	26
Figura 13 – Código SQL Correspondente alínea g)	26
Figura 14 – Índice	27
Figura 15 – Base de Dados implementada em MongoDB	29
Figura 16 – Query em MongoDB alínea a)	29
Figura 17 – Query em MongoDB alínea b)	30
Figura 18 – Query em MongoDB alínea c)	30
Figura 19 – Query em MongoDB alínea d)	31

## Índice de Tabelas

Tabela 1 – Identificação e caracterização das entidades	6
Tabela 2 – Relacionamentos das entidades	7
Tabela 3 – Atributos da Entidade Cliente	8
Tabela 4 – Atributos da Entidade CP	9
Tabela 5 – Atributos do relacionamento “aquisição”	10
Tabela 6 – Atributos do relacionamento “aquisição”	10
Tabela 7 – Atributos da Entidade Biblioteca	11
Tabela 8 – Atributos da Entidade Filme	12
Tabela 9 – Atributos da Entidade Série	13
Tabela 10 – Atributos da Entidade Produção	14

# **1. Introdução**

## **1.1. Contextualização**

O Sr. Paulo é o dono de um cinema no Braga Parque em Braga. Com o agravamento da doença Covid-19, o Sr. Paulo deparou-se com um problema quando o governo decretou confinamento geral. Foi forçado a fechar o seu negócio e a mandar todos os seus empregados para casa. Com o estabelecimento fechado e os apoios do estado insuficientes para pagar os salários, renda e despesas, o Sr. Paulo não queria recorrer a despedimento coletivo, logo teve a ideia de criar um serviço streaming. Assim surgiu a empresa Deixa Ver.

O Sr. Paulo decidiu então investir na implementação de uma base de dados que o permitiria armazenar e controlar todos os serviços prestados pela empresa.

## **1.2. Apresentação do Caso de Estudo**

Face à situação epidémica que enfrentámos nos últimos dois anos, muitos negócios viram-se forçados a optar por uma via online do seu planeamento, de forma a conseguirem, pelo menos, sobreviver à situação em que todos nos encontramos.

De forma a fazer corretamente esta adaptação a uma realidade diferente, é necessário a criação de uma base de dados segura e consistente, de forma a dar possibilidade aos clientes de assistir ao mesmo tipo de conteúdo que assistiriam no cinema do Sr. Paulo, mas no conforto da sua casa ou a partir de qualquer lugar que desejem.



### **1.3. Motivação e Objetivos**

No caso do Sr. Paulo a motivação do mesmo para esta mudança é evidente, visto que se viu restringido de vender um único bilhete durante meses, tendo então de mudar o seu modelo de negócio de forma que este sobrevivesse.

Com a criação da base de dados para a nova plataforma online, alguns requisitos são extremamente importantes para o pleno funcionamento do negócio. Ao guardar dados como o género do filme ou série e a data da compra/aluguer, é possível saber qual o género mais popular em que altura do ano, trazendo assim mais variedade de filmes e séries desse género nessa altura, de forma a maximizar o lucro.

Analisando as compras de cada cliente, é possível também propor ao cliente filmes semelhantes ao que este está habituado a comprar, por email ou pelo seu contacto telefónico. Seguindo a análise destes dados bastante importantes, é possível gerir o stock da variedade de conteúdo disponível, de maneira a controlar o número de transações, o tipo destas transações e o conteúdo das mesmas.

### **1.4. Estrutura do Relatório**

Com este relatório pretendemos demonstrar o pensamento e processo de criação de uma base de dados, no contexto anteriormente descrito. No segundo capítulo é feita um levantamento dos requisitos necessários para a base de dados, bem como as soluções encontradas para satisfazer os mesmos.

No capítulo 3 é feita a Modelação Conceptual, onde são identificados e descritos os vários relacionamentos, entidades e atributos, bem como os seus propósitos. Também nesse capítulo, é possível observar o Diagrama ER.

Por fim, é feito um ponto de situação do projeto, concluindo o que vem a ser feito neste projeto até ao momento.

## **2. Levantamento e Análise de Requisitos**

A componente base do projeto é a análise dos requisitos, quando esta é feita corretamente, tendo requisitos bem estruturados e concretos de forma a garantir as necessidades do projeto em questão, é um grande passo para a correta execução do projeto.

### **2.1. Método de levantamento e de análise de requisitos**

Após conversarmos com o Sr. Paulo, o grupo de trabalho apresentou uma proposta de requisitos, após uma análise crítica do seu plano de negócio, de forma que esta fosse eficiente e a mais correta possível.

### **2.2. Requisitos Levantados**

Começamos então o levantamento de requisitos, os quais dividimos em 3 categorias. Os requisitos de descrição, onde apresentamos os requisitos necessários para o funcionamento e armazenamento dos dados, os requisitos de exploração imprescindíveis para a apresentação e listagem de dados e os requisitos de controlo, com o objetivo de proteger e encapsular as informações dependendo do tipo de acesso à base de dados.

#### **2.2.1 Requisitos de descrição**

- A base de dados deve conter as aquisições efetuadas, com o id da aquisição, a data da mesma, o valor, preço e quantidade

- No caso dos alugueres, diferem apenas na data final da aquisição, visto que no caso da compra é nula, mas no caso do aluguer tem uma data limite
- Deve também guardar o login do cliente, juntamente com o seu nome, NIF, contacto com o número e tipo de contacto, ou seja, se é telemóvel ou fixo, e código postal, o qual terá o código e a localidade
- Cada filme terá o seu ID, a duração em horas, a classificação, a sinopse, a data de lançamento, a classificação etária recomendada, o género, o título e o preço
- Quanto às séries cada uma terá um ID, o número de episódios e temporadas, a duração média de cada episódio, a classificação, a sinopse, a data de lançamento, a classificação etária recomendada, o género, o título e o preço
- Tanto os filmes como as séries têm uma produção, e esta entidade terá a realização, o elenco e a produtora
- Cada cliente tem uma biblioteca com filmes e/ou séries

### **2.2.2 Requisitos de exploração**

- Apresentar todos os clientes, filmes e séries
- Apresentar as compras de um dado cliente
- Apresentar o top 3 filmes mais comprados
- Apresentar a produção de uma série
- Apresentar as compras num dado dia

### **2.2.3 Requisitos de controlo**

- O Sr. Paulo é o único responsável pela empresa, logo terá permissão para visualizar toda e qualquer informação presente na base de dados. Terá ainda permissão para adicionar novos filmes, séries, clientes, compras e alugueres. Terá ainda permissão para utilizar todas as funções desenvolvidas

## **2.3 Análise e validação geral dos requisitos**

O grupo prevê um bom funcionamento da base de dados, de acordo com os requisitos que foram levantados, de forma a atender todas as necessidades do negócio do Sr.Paulo. Procuramos também proteger os seus clientes, fazendo diferentes logins para que os utilizadores possam ter a sua conta particular.

## **3. Modelo Conceptual**

Nesta parte apresentamos a modelação conceptual, escrita em notação CHEN, utilizando a ferramenta brModelo, para fazermos uma representação clara e precisa dos seus diversos componentes, como: entidade, relacionamentos, atributos e chaves.

### **3.1 Apresentação da abordagem de modelização realizada**

Tendo em conta os requisitos elaborados no capítulo anterior, tentamos desenvolver um modelo conceptual. Este modelo consiste no uso de entidades, em que cada uma das entidades será caracterizada pelos seus atributos. É necessário também uma peça que faça a ligação entre as entidades, neste caso serão os relacionamentos

### 3.2 Identificação e caracterização das entidades

Entidades	Descrição	Ocorrências
Cliente	Representação de quem usa a Deixa Ver como seu serviço de streaming.	Cada cliente tem um código postal, um histórico de compras, pode ter zero ou mais alugueres e pode fazer zero ou mais compras e alugueres.
Biblioteca	Entidade que guarda todos os filmes e séries comprados por um cliente.	Cada biblioteca tem um e um só cliente associado.
CP	Representa o Código Postal e localidade de um cliente.	Cada CP tem zero ou mais clientes associados.
Filme	Filme disponível para compra ou aluguer no serviço de streaming Deixa Ver.	Cada Filme pode ser comprado ou alugado zero ou mais vezes e tem uma equipa de produção.
Série	Série disponível para compra ou aluguer no serviço de streaming Deixa Ver.	Cada Série pode ser comprada ou alugada zero ou mais vezes e tem uma equipa de produção.

Produção	Representação da equipa de produção de um filme ou série	Cada equipa de produção tem um ou mais filmes e/ou séries associados.
----------	--	--

**Tabela 1 – Identificação e caracterização das entidades.**

### **3.3 Identificação e caracterização dos relacionamentos**

Entidade 1	Multiplicidade 1	Relacionamento	Entidade 2	Multiplicidade 2
Cliente	1-1	tem	Biblioteca	1-1
Cliente	0-n	tem	CP	1-1
Cliente	0-n	Aquisição	Filme	0-n
Cliente	0-n	Aquisição	Série	0-n
Filme	1-n	tem	Produção	1-1
Série	1-n	tem	Produção	1-1

**Tabela 2 – Relacionamentos das entidades.**

### 3.4 Identificação e Caraterização da associação dos Atributos com as Entidades e Relacionamentos

Atributos	Descrição	Tipo de dados	Multivalor	Null
Email	Email do cliente	VARCHAR(50)	Não	Não
Password	Password associada á conta	VARCHAR(50)	Não	Não
Nome	Nome do cliente	VARCHAR(50)	Não	Não
NIF	Número de identificação fiscal	CHAR(9)	Não	Não
Contacto	Contacto associado ao cliente, seja telemóvel ou telefone	INT	Não	Não

**Tabela 3 – Atributos da Entidade Cliente.**

Atributos	Descrição	Tipo de dados	Multivalor	Null
Id	Número de identificação do código postal	INT	Não	Não
Nome	Nome da localidade de residência do cliente	VARCHAR(50)	Não	Não
codigoPostal	Código postal da morada da residência do cliente	CHAR(8)	Não	Não

**Tabela 4 – Atributos da Entidade CP.**



Atributos	Descrição	Tipo de dados	Multivalor	Null
Id	Número de identificação da aquisição	INT	Não	Não
Preço	Preço total da compra/aluguer	FLOAT	Não	Não
data	Data de aquisição do filme/série	DATE	Não	Não
dataFim	Em caso de aluguer, data em que termina o mesmo	DATE	Não	Sim

**Tabela 5 – Atributos do relacionamento “aquisição”.**

Atributos	Descrição	Tipo de dados	Multivalor	Null
Id	Número de identificação da aquisição	INT	Não	Não
Preço	Preço total da compra/aluguer	FLOAT	Não	Não
data	Data de aquisição do filme/série	DATE	Não	Não
dataFim	Em caso de aluguer, data em que termina o mesmo	DATE	Não	Sim

**Tabela 6 – Atributos do relacionamento “aquisição”.**

Atributos	Descrição	Tipo de dados	Multivalor	Null
Id	Número de identificação da biblioteca	INT	Não	Não

**Tabela 7 – Atributos da Entidade Biblioteca**

Atributos	Descrição	Tipo de dados	Multivalor	Null
Id	Número de identificação do filme	INT	Não	Não
Título	Nome do filme	VARCHAR(50)	Não	Não
Classificação	Classificação do filme	FLOAT	Não	Não
ClassificaçãoEtária	Informação sobre a faixa etária para a qual os filmes não se recomendam	VARCHAR(4)	Não	Não
Duração	Tempo de duração do filme	VARCHAR(8)	Não	Não
Gênero	Gênero do filme, utilizado para distinguir variados tipos de filme	VARCHAR(50)	Não	Não
Sinopse	Breve descrição sobre o filme	VARCHAR(200)	Não	Não
dataLançamento	Data de estreia do filme	DATE	Não	Não
Preço	Preço do filme	FLOAT	Não	Não

**Tabela 8 – Atributos da Entidade Filme.**

Atributos	Descrição	Tipo de dados	Multivalor	Null
Id	Número de identificação da série	INT	Não	Não
Título	Nome da série	VARCHAR(50)	Não	Não
Classificação	Classificação da série	FLOAT	Não	Não
ClassificaçãoEtária	Informação sobre a faixa etária para a qual a série não se recomenda	VARCHAR(4)	Não	Não
Duração	Tempo de duração total da série	VARCHAR(8)	Não	Não
Gênero	Gênero da série, utilizado para distinguir variados tipos de séries	VARCHAR(50)	Não	Não
Sinopse	Breve descrição sobre a série	VARCHAR(200)	Não	Não
dataLançamento	Data de estreia da série	DATE	Não	Não
Preço	Preço da série	FLOAT	Não	Não
Ep	Número de episódios	INT	Não	Não
Temporada	Número de temporadas	INT	Não	Não

**Tabela 9 – Atributos da Entidade Série.**

Atributos	Descrição	Tipo de dados	Multivalor	Null
Id	Número de identificação da produção	INT	Não	Não
Direção	Equipa responsável pela direção e realização do filme	VARCHAR(50)	Não	Não
Elenco	Principal elenco que participa no respetivo filme/série	VARCHAR(50)	Não	Não
Produtora	Empresa responsável pela produção do filme/série	VARCHAR(50)	Não	Não

**Tabela 10 – Atributos da Entidade Produção.**

### 3.5 Detalhe ou generalização de entidades

Aqui apresentamos uma explicação para a necessidade de cada uma das entidades identificadas neste trabalho:

- **Cliente:** Se devido as restrições da covid-19 o Sr. Paulo não tivesse de fechar as portas aos seus clientes este não precisaria de financiar a empresa Deixa Ver, e, portanto, com o seu estabelecimento fechado e vários utilizadores a aderir aos serviços de streaming esta entidade é obviamente necessária.
- **CP:** Devido à possibilidade de vários clientes poderem possuir o mesmo código postal e localidade decidimos criar uma entidade para facilitar a gestão invés de retermos vária informação repetida
- **Filme:** Como a empresa Deixa Ver, necessita obviamente de um sistema de gestão dos filmes a apresentar aos seus clientes, então terá de existir a entidade filmes, já que a empresa tem uma vasta biblioteca a apresentar.
- **Série:** Tal como filmes a empresa Deixa Ver também apresenta séries aos seus clientes, portanto, necessita também de um sistema de gestão das séries a apresentar, então terá de existir a entidade séries, já que a empresa tem uma vasta biblioteca a apresentar.
- **Aluguer:** Como a empresa Deixa Ver permite aos seus utilizadores alugarem um filme ou série terá também de existir a entidade Aluguer para que seja possível fazer uma gestão de todos os alugueres dos vários clientes.
- **Aquisição:** A empresa Deixa Ver permite a compra e aluguer do seu conteúdo de streaming, portanto é trivial que vai ser necessário um sistema para a gestão das compras e alugueres dos vários utilizadores, logo a entidade compra vai ser obviamente necessária.
- **Produção:** Como uma equipa de produção registada na Deixa Ver pode estar associada a uma ou mais séries ou filmes achamos necessário criarmos uma entidade que faça a gestão destas, de maneira a não retermos informação repetida.

## 3.6 Diagrama ER

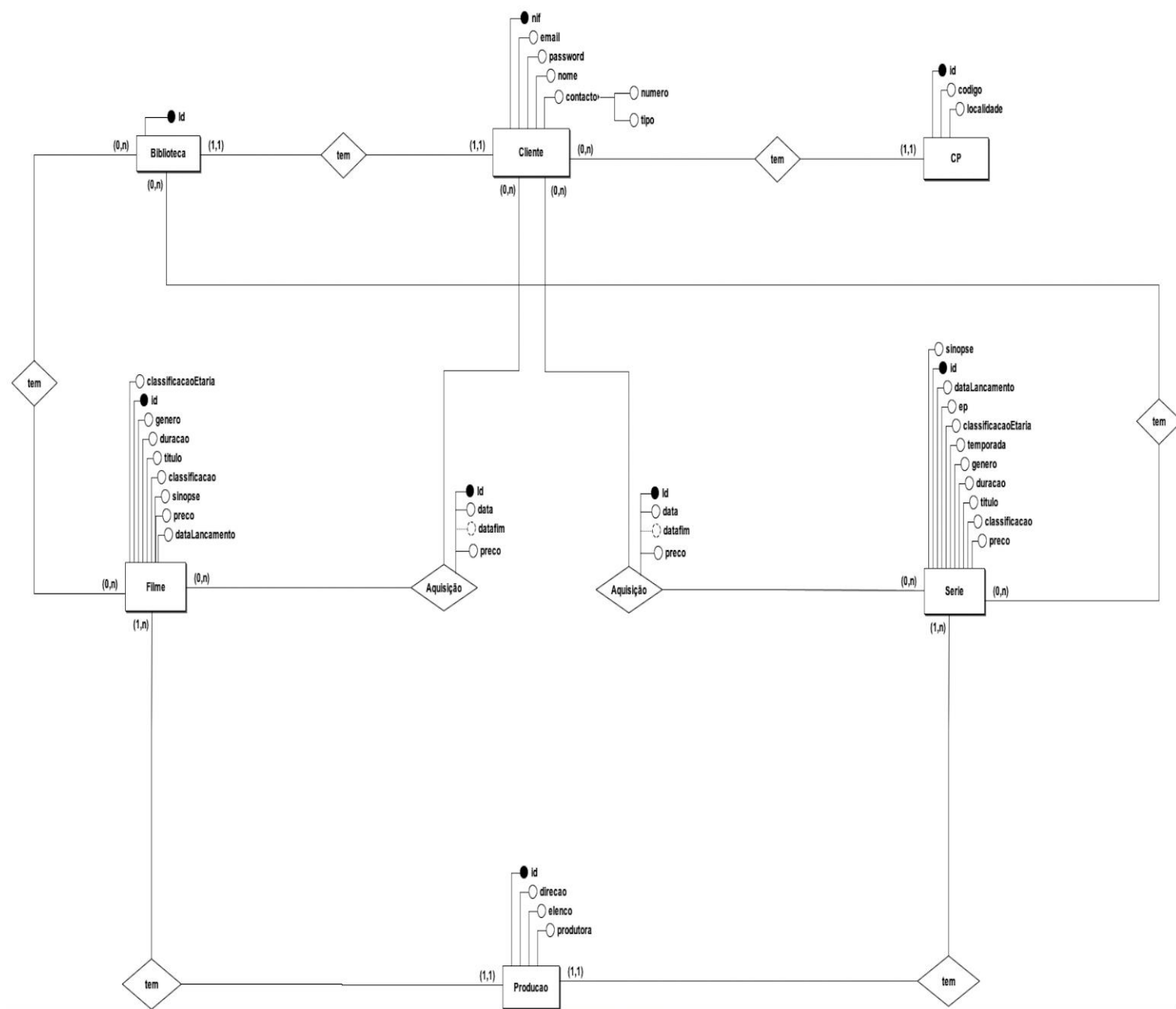


Figura 1 - Diagrama ER

## 4. Modelação Lógica

### 4.1 Construção e validação do modelo de dados lógico

O modelo lógico foi construído a partir do nosso modelo conceptual, como se pode observar foram feitas algumas alterações como por exemplo, juntamos as tabelas Filme e Série numa só tabela denominada Conteúdo, assim como removemos a entidade biblioteca e separamos as informações de um filme ou série numa única entidade denominada Descrição, estas alterações serão justificadas mais a frente no relatório na secção 6.

Neste modelo as entidades e os seus relacionamentos são convertidos em tabelas e os identificadores de uma entidade passam a designar-se por Primary Key, ou Foreign Key no caso de o identificador de uma entidade se encontrar noutra sobre forma de tabela.

#### 1. Cliente:

**1.1. Primary Key:** NIF como INT

**1.2. Atributos:** email: VARCHAR(50), password: VARCHAR(50), nome: VARCHAR(50)

**1.3. Foreign Key:** CP\_id INT

#### 2. Aquisição:

**2.1. Primary Key:** id como INT

**2.2. Atributos:** tipoAquisição: VARCHAR(45), data: DATE, nrDias: INT

**2.3. Foreign Key:** Conteudo\_id INT, Cliente\_nif INT



### 3. Conteúdo

**3.1. Primary Key:** id como INT

**3.2. Atributos:** tipo: VARCHAR(45), titulo: VARCHAR(45),  
preçoCompra: DECIMAL(6,2), PreçoAluguer: DECIMAL(6,2)

**3.3. Foreign Key:** Producao\_id INT, Descricao\_nif INT

### 4. Descrição

**4.1. Primary Key:** id como INT

**4.2. Atributos:** classificacaoEtaria: VARCHAR(4), duracaoFilme: TIME,  
classificação: DECIMAL(8,2), sinopse: TEXT, dataLançamento:  
DATE, nrTemporadas: INT

**4.3. Foreign Key:** Não possui

### 5. Producao

**5.1. Primary Key:** id como INT

**5.2. Atributos:** direcao: VARCHAR(45), produtora: VARCHAR(45),  
emissora: VARCHAR(45)

**5.3. Foreign Key:** Não possui

### 6. Contacto

**6.1. Primary Key:** numero como INT

**6.2. Atributos:** Não possui

**6.3. Foreign Key:** Cliente\_nif INT

### 7. CP

**7.1. Primary Key:** id como INT

**7.2. Atributos:** código: CHAR(8), localidade: VARCHAR(50)

**7.3. Foreign Key:** Não possui

## 4.2 Desenho do modelo lógico

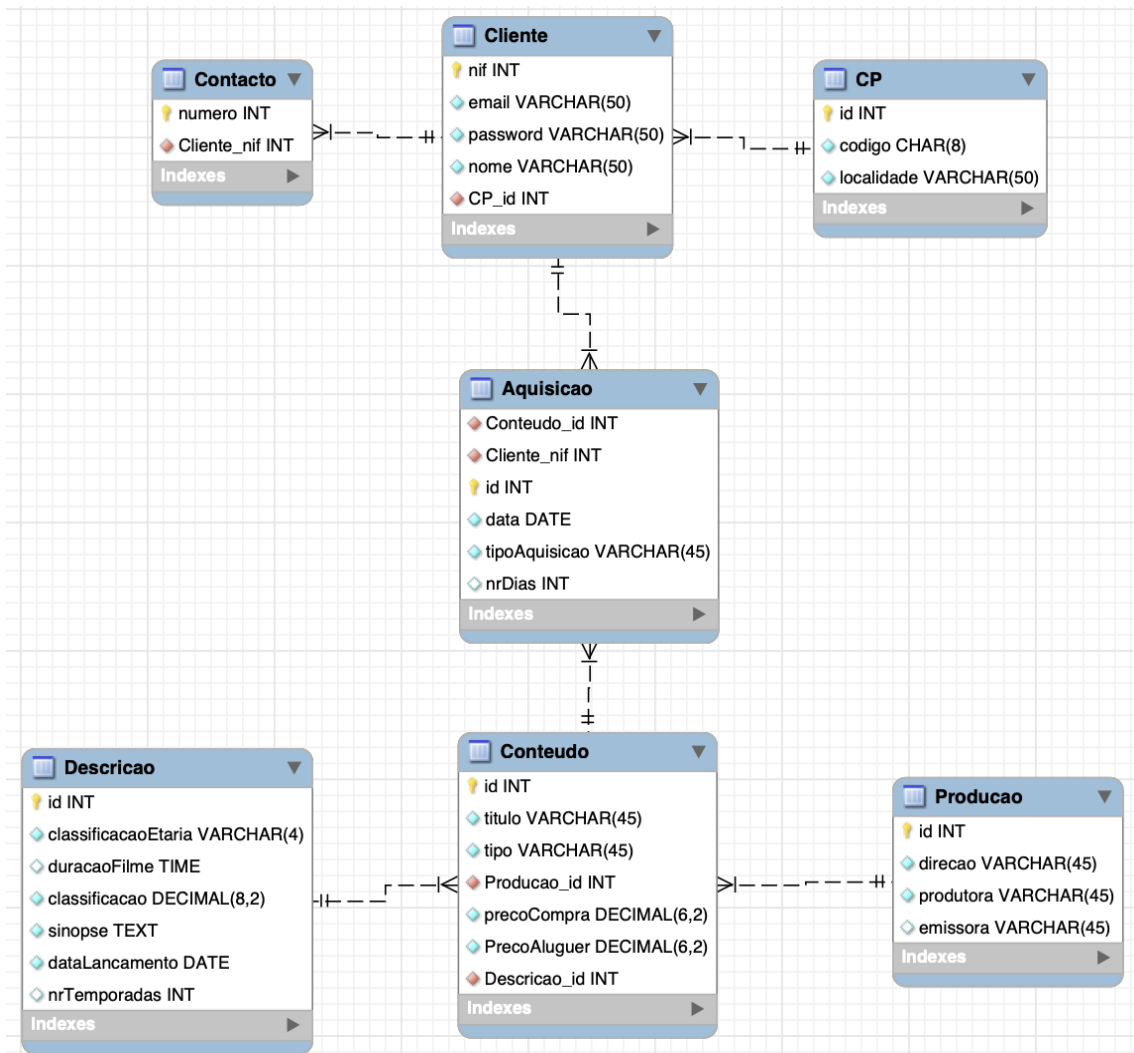


Figura 2 - Modelo Lógico

### 4.3 Validação do modelo através da normalização

A normalização é um conjunto de regras (1FN, 2FN, 3FN), que pretende evitar a redundância de dados e aumentar o desempenho do modelo. Estas regras consistem na examinação de atributos de uma entidade e das relações entre entidades e têm como principal objetivo evitar anomalias na inclusão, exclusão e alteração de registos.

Para verificarmos a normalização do modelo temos que verificar se o mesmo satisfaz as 3 fórmulas normais:

- Primeira Fórmula Normal – A 1FN diz-nos que os atributos precisam de ser atómicos, ou seja, as tabelas não podem ter valores repetidos nem pode existir atributos multivalorados. Com uma simples verificação do nosso modelo podemos afirmar que tal acontece, logo a 1FN é válida;
- Segunda Fórmula Normal – A 2FN diz-nos que só pode ser cumprida se a 1FN for satisfeita, e que os atributos normais devem depender unicamente da chave primária da tabela. Como exemplo podemos usar a password do cliente, este atributo depende unicamente da chave primária, o NIF. O mesmo é observável no resto do modelo, logo a 2FN é válida;
- Terceira Fórmula Normal – A 3FN diz-nos que após serem válidas as duas fórmulas anteriores, devemos verificar que todos os atributos de uma tabela devem ser independentes uns dos outros. Podemos verificar de facto que o nosso modelo não possui nenhum atributo proporcional a outro ou possível ser gerado a partir de outro. Provando que a 3FN é válida.

Como podemos verificar no modelo lógico apresentado no capítulo 4.2, este verifica as 3 fórmulas normais, logo podemos afirmar que o modelo é válido em termos de normalização.

#### 4.4 Validação do modelo com interrogações do utilizador

Para validar o nosso modelo lógico desenvolvido selecionamos algumas interrogações dos requisitos de exploração e, usando álgebra relacional, mostramos de que forma o modelo as satisfaz.

a) Histórico do cliente através do NIF

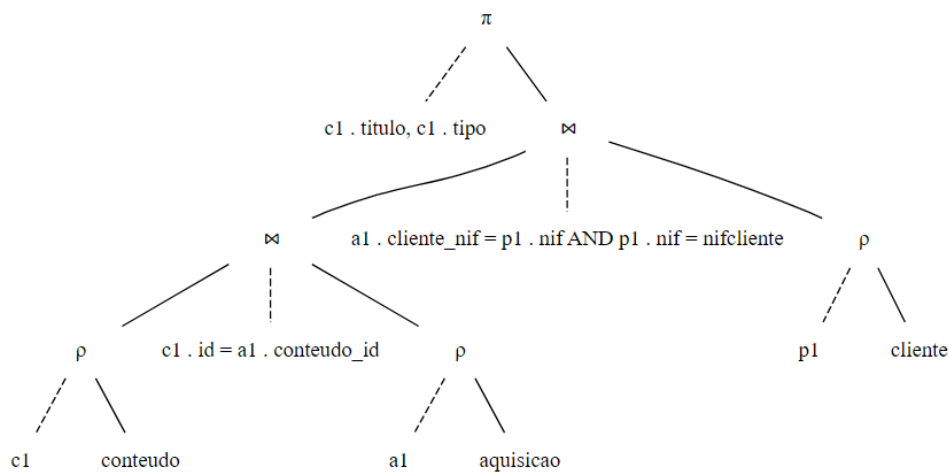


Figura 3 - Árvore histórico do cliente através do NIF

Primeiro selecionamos o título e tipo do conteúdo, de seguida agrupamos a tabela aquisição pelos id's, fazendo o mesmo para a tabela cliente, mas desta vez com os NIFs. Da tabela cliente escolhemos apenas os registos cujo o atributo nif é igual ao nif dado como input.

b) Obter produção através do título

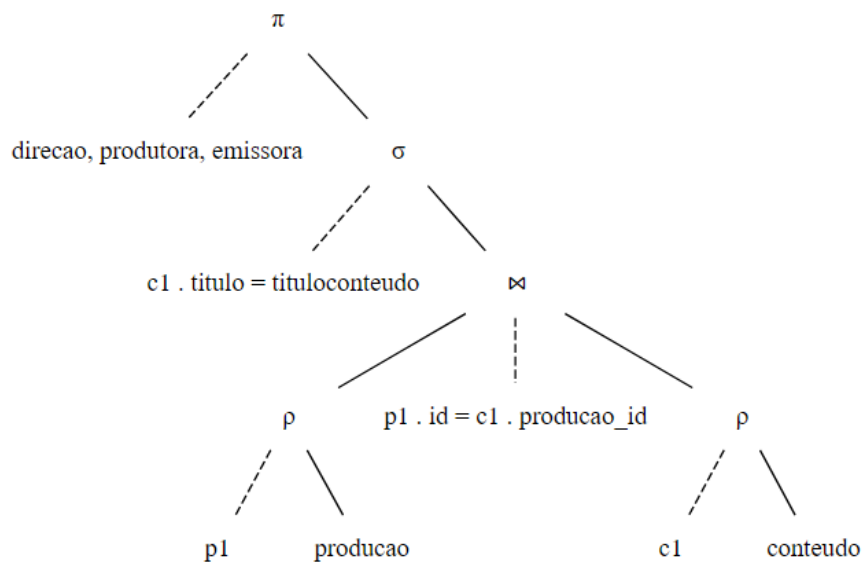


Figura 4 - Árvore obter produção através do título

Começamos por selecionar a direção, produtora e emissora da tabela produção. De seguida agrupamos a tabela conteúdo pelos id's. Por fim escolhemos da tabela conteúdo os registos cujo atributo titulo é igual ao titulo dado como input.

c) Compras numa dada data

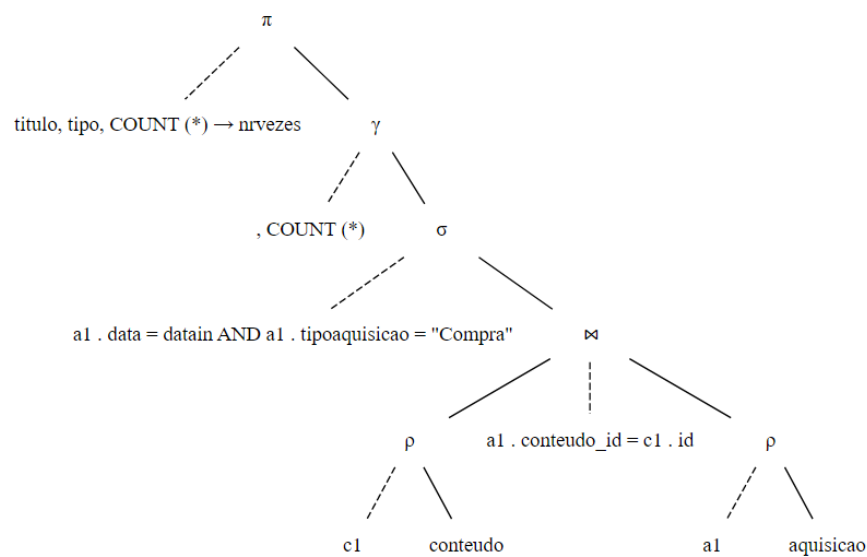


Figura 5 - Árvore compras numa dada data

Selecionamos o título e tipo da tabela conteúdo. Após esse passo agrupamos a tabela aquisição pelos id's. Da tabela aquisição escolhemos apenas os registos onde as datas são iguais às dadas como input e o tipo de aquisição sendo uma compra. Finalmente, fazemos count às vezes que um título aparece.

d) Top N Séries/Filmes mais adquiridos

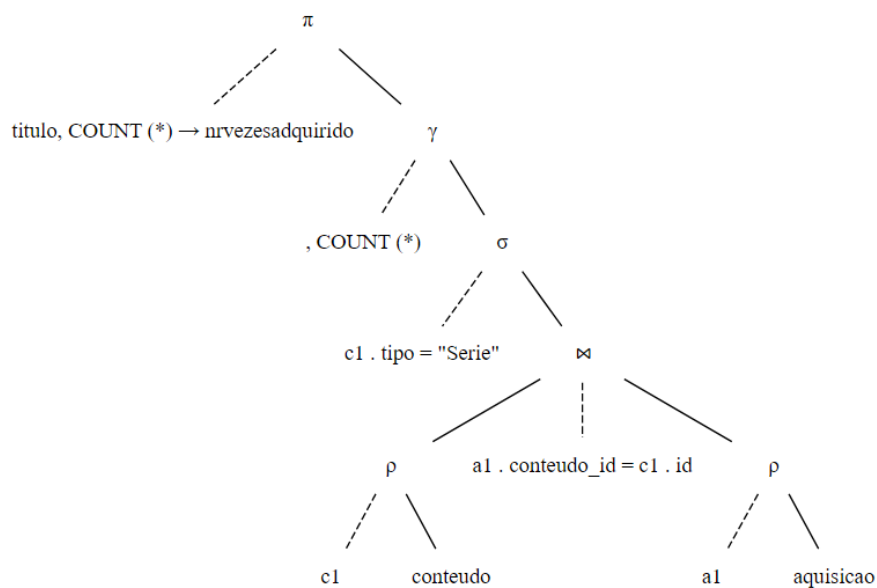


Figura 6 - Árvore Top N séries/filmes

Começamos por selecionar o título da tabela conteúdo. Agrupamos a tabela aquisição, nos casos em que os id's são iguais. De seguida, da tabela conteúdo escolhemos apenas os registos cujo atributo tipo é igual a Série. No caso do Top N Filmes, a árvore é bastante idêntica, mudando apenas a última etapa em que o atributo chave é ser Filme ao invés de Série.

## 4.5 Conjunto de procedimentos em MySQL

A partir das árvores geradas na secção anterior fomos capazes de converter as interrogações do utilizador para SQL.

### a) Ver conteudo

```
DELIMITER $$
CREATE PROCEDURE histCliente(IN nifCliente VARCHAR(200))
BEGIN
SELECT C1.titulo,C1.tipo from Conteudo C1
INNER JOIN Aquisicao A1 on C1.id = A1.Conteudo_id
INNER JOIN Cliente P1 on A1.Cliente_nif = P1.nif AND P1.nif=nifCliente
ORDER BY tipo;
END $$
```

Figura 7 - Código SQL Correspondente alínea a)

### b) Dado um titulo de um conteudo ver a sua produção

```
DELIMITER $$
CREATE PROCEDURE conteudoProd(IN tituloConteudo VARCHAR(200))
BEGIN
SELECT direcao,produtora,emissora from Producao P1
INNER JOIN Conteudo C1 on P1.id = C1.Producao_id
Where C1.titulo = tituloConteudo;
END $$

CALL conteudoProd('Wolf of Wall Street');
CALL conteudoProd('The Batman');
CALL conteudoProd('Chuck');
```

Figura 8 - Código SQL Correspondente alínea b)

**c) Compras numa dada data**

```
DELIMITER $$
CREATE PROCEDURE comprasDias(IN dataIN VARCHAR(200))
BEGIN
SELECT titulo,tipo,COUNT(*) as nrVezes from Conteudo C1
INNER JOIN Aquisicao A1 on A1.Conteudo_id = C1.id
WHERE A1.data = dataIN AND A1.tipoAquisicao = 'Compra'
GROUP BY titulo
ORDER BY nrVezes DESC;
END $$
```

Figura 9 - Código SQL Correspondente alínea c)

**d) Compras e alugueres num dado ano**

```
DELIMITER $$
CREATE PROCEDURE listaAno(IN anoIN VARCHAR(200))
BEGIN
SELECT titulo,tipo,COUNT(*) as nrVezes from Conteudo C1
INNER JOIN Aquisicao A1 on A1.Conteudo_id = C1.id
WHERE YEAR(A1.data) = anoIN AND A1.tipoAquisicao in ('Compra','Aluguer')
GROUP BY titulo
ORDER BY nrVezes DESC;
END $$
```

Figura 10 - Código SQL Correspondente alínea d)

**e) Top N filmes do serviço (compra e aluguer)**

```
DELIMITER $$
CREATE PROCEDURE topNFilmes(IN nIN INT)
BEGIN
SELECT titulo,COUNT(*) as nrVezesAdquirido from Conteudo C1
INNER JOIN Aquisicao A1 on A1.Conteudo_id = C1.id
WHERE C1.tipo = 'Filme'
GROUP BY titulo
ORDER BY nrVezesAdquirido DESC LIMIT nIN;
END $$
```

Figura 11 - Código SQL Correspondente alínea e)



**f) Top N séries do serviço (compra e aluguer)**

```
DELIMITER $$
CREATE PROCEDURE topNSeries(IN nIN INT)
BEGIN
SELECT titulo,COUNT(*) as nrVezesAdquirido from Conteudo C1
INNER JOIN Aquisicao A1 on A1.Conteudo_id = C1.id
WHERE C1.tipo = 'Serie'
GROUP BY titulo
ORDER BY nrVezesAdquirido DESC LIMIT nIN;
END $$
```

Figura 12 - Código SQL Correspondente alínea f)

**g) Quanto gastou um cliente no serviço dado seu nif**

```
DELIMITER $$
CREATE PROCEDURE gastoCliente(IN nifCliente VARCHAR(200))
BEGIN
SELECT nifCliente,COALESCE(SUM(precoCompra)) + COALESCE(SUM(precoAluguer)) as valorTotal from Conteudo
WHERE id IN (SELECT Conteudo_id from Aquisicao A1
INNER JOIN Cliente C1 on A1.Cliente_nif = C1.nif
WHERE C1.nif = nifCliente);
END $$
```

Figura 13 - Código SQL Correspondente alínea g)

## 4.6 Escolha, definição e caracterização de índices em SQL

Ao criar um objeto, são automaticamente criados índices para as colunas com restrição PRIMARY KEY , chamados índices implícitos. Porém, nós também podemos definir os nossos próprios índices, ou seja, índices explícitos. A vantagem de criar índices é que estes aceleram operações de seleção e consulta, mas por outro lado abrandam operações de inserção e atualização.

Reparamos nas nossas queries que a operação de seleção que mais fazíamos era a seleção do título de um conteudo, logo achamos boa escolha definir um índice para a coluna título da tabela Conteudo.

```
CREATE INDEX tituloIN ON Conteudo(titulo);  
  
SELECT * FROM Conteudo where titulo = 'Wolf of Wall Street';
```

Figura 14 - Índice

## **5. MongoDB**

### **5.1 Migração dos dados do sistema relacional para o sistema de dados em MongoDB**

Como ambiente de trabalho, o grupo decidiu trabalhar com o MongoDB compass, pois achamos que a sua interface é limpa e organizada e é fácil de ver as collections e os seus respetivos dados.

Para migrar então os nossos dados do sistema relacional para MongoDB vamos primeiro usar o Table Data Export Wizard do MySQL para exportar os dados em formato JSON. De seguida, criamos uma base de dados nova no MongoDB e as respetivas collections, nomeadamente Aquisição, Cliente, Contacto, Conteúdo, CP, Descrição e Produção, e após isso importamos os respetivos ficheiros JSON com os dados. E ficamos assim com a nossa base de dados implementada no sistema NoSQL MongoDB.

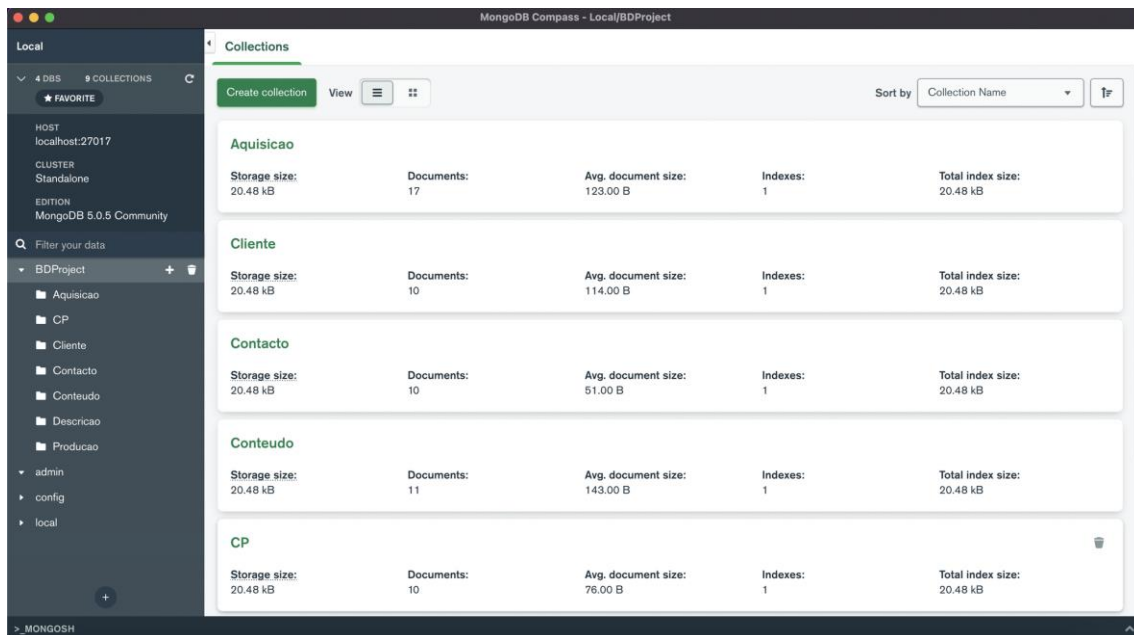


Figura 15 - Base de Dados implementada em MongoDB

## 5.2 Conjunto de queries em MongoDB

Implementamos em MongoDB um conjunto de queries equivalente às que apresentamos na secção 4.4 para mostrar a operacionalidade do sistema.

- a) Ver todas as compras de um cliente dado o seu nif

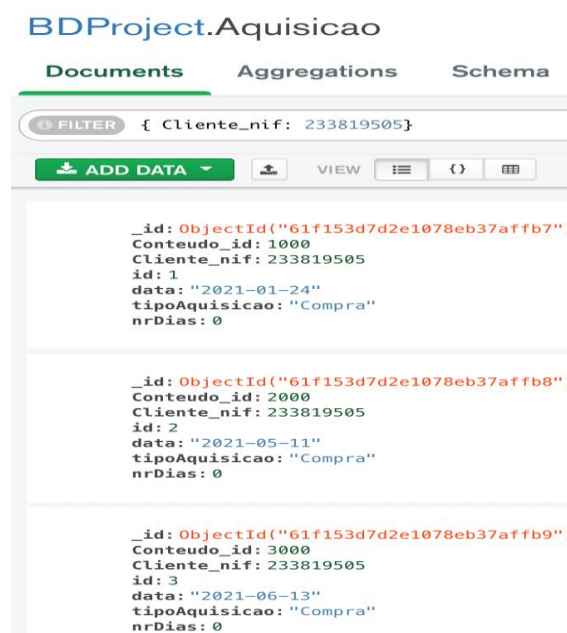


Figura 16 - Query em MongoDB alínea a)

b) Ver compras num dado dia

The screenshot shows the MongoDB interface for the 'BDProject.Aquisicao' database. The 'Documents' tab is selected. A filter is applied: `{data: "2014-01-09"}`. The results show three documents, all with `data: "2014-01-09"` and `tipoAquisicao: "Compra"`.

```
{ "_id": ObjectId("61f153d7d2e1078eb37affc3"),
  "Conteudo_id": 1000,
  "Cliente_nif": 116105232,
  "id": 13,
  "data": "2014-01-09",
  "tipoAquisicao": "Compra",
  "nrDias": 0 }
{ "_id": ObjectId("61f153d7d2e1078eb37affc4"),
  "Conteudo_id": 1000,
  "Cliente_nif": 293514046,
  "id": 14,
  "data": "2014-01-09",
  "tipoAquisicao": "Compra",
  "nrDias": 0 }
{ "_id": ObjectId("61f153d7d2e1078eb37affc5"),
  "Conteudo_id": 1000,
  "Cliente_nif": 266084141,
  "id": 15,
  "data": "2014-01-09",
  "tipoAquisicao": "Compra",
  "nrDias": 0 }
```

Figura 17 - Query em MongoDB alínea b)

c) Ver compras num dado ano

The screenshot shows the MongoDB interface for the 'BDProject.Aquisicao' database. The 'Documents' tab is selected. A filter is applied: `{data: { $gte: ("2021-01-01"), $lte: ("2021-31-21") }}`. The results show three documents, all with `data` in 2021 and `tipoAquisicao: "Compra"`.

```
{ "_id": ObjectId("61f153d7d2e1078eb37affb7"),
  "Conteudo_id": 1000,
  "Cliente_nif": 233819505,
  "id": 1,
  "data": "2021-01-24",
  "tipoAquisicao": "Compra",
  "nrDias": 0 }
{ "_id": ObjectId("61f153d7d2e1078eb37affb8"),
  "Conteudo_id": 2000,
  "Cliente_nif": 233819505,
  "id": 2,
  "data": "2021-05-11",
  "tipoAquisicao": "Compra",
  "nrDias": 0 }
{ "_id": ObjectId("61f153d7d2e1078eb37affb9"),
  "Conteudo_id": 3000,
  "Cliente_nif": 233819505,
  "id": 3,
  "data": "2021-06-13",
  "tipoAquisicao": "Compra",
  "nrDias": 0 }
```

Figura 18 - Query em MongoDB alínea c)

d) Ver todos os clientes registrados no serviço

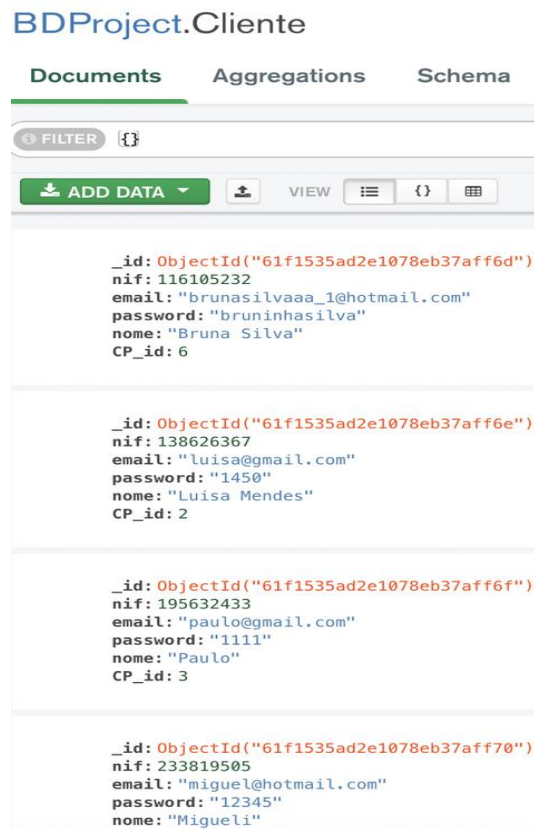


Figura 19 - Query em MongoDB alínea d)

## **6. Análise crítica do trabalho realizado e Conclusão**

Depois de efetuada a primeira parte do projeto, fomos críticos e honestos perante a primeira parte apresentada, reconhecendo que haveria pontos em que seria possível melhorar e acabámos então por efetuar algumas alterações de forma a melhorar a coesão e eficiência da base de dados.

Após efetuarmos a modelação lógica, notámos algumas redundâncias no trabalho. Um dos casos foi a distinção entre aluguer e compra. Notámos que estas duas operações eram bastante semelhantes, mesmo em termos de atributos, sendo apenas distinguíveis na questão do tempo de aluguer, que no caso da compra se pode considerar infinito. Assim, optámos por juntar estas duas operações numa só, a aquisição.

Outro exemplo que não foi considerado inicialmente foi a biblioteca. Com o decorrer do trabalho deparámo-nos que a utilidade da mesma seria nula, visto que com uma simples query conseguiríamos obter a biblioteca de um cliente, não havendo assim a necessidade de uma entidade específica para isso.

Por fim, à semelhança do caso da aquisição, também no caso dos filmes e séries deparámo-nos com as suas vastas semelhanças. Assim, acabámos por juntar ambas as entidades numa só, o conteúdo, separando ainda a sua descrição e produção em duas entidades diferentes, de forma a normalizar o projeto.

Após estas alterações, fomos validando o modelo, de forma a garantir que todos os requisitos inicialmente propostos continuariam a ser cumpridos integralmente. Concluímos então que as formas de normalização foram cumpridas, tanto como os requisitos, sentindo uma sensação de satisfação acerca do trabalho produzido.

A gestão eficiente da base de dados foi uma prioridade, sendo mais uma das razões para as alterações efetuadas, tentando ao máximo evitar redundâncias e escolhendo os atributos mais adequados, agrupando-os nas situações em que fazia sentido tal acontecer.

Em relação a um trabalho futuro, reconhecemos que existem alguns pontos a melhorar. Com o crescimento gradual da base de dados, a organização dos conteúdos tornar-se-iam mais difíceis, havendo a necessidade de reforçar esse aspeto, otimizando a forma como esses conteúdos seriam guardados.

Para concluir, sentimo-nos realizados com o trabalho que foi feito, havendo a sensação de satisfação devido ao facto das exigências inicialmente propostas terem sido cumpridas com sucesso.

Por fim, gostaríamos de agradecer aos Professores Orlando Belo e António Abelha pelo tempo disponibilizado em relação às questões colocadas ao longo do semestre.



## 7. Bibliografica

- MongoDB Query Documents. <<https://docs.mongodb.com/manual/tutorial/query-documents/>>
- StackOverflow. < <https://stackoverflow.com/>>
- Fichas Práticas da cadeira Base de Dados disponiveis na BlackBoard.
- W3Schools. < <https://www.w3schools.com/sql/>>
- Database Systems A Practical Approach to Design, Implementation, and Management Global Edition by Thomas Connolly, Carolyn Begg (z-lib.org)
- DelftStack How to SQL. < <https://www.delftstack.com/howto/mysql/>>