



Universidade do Minho

Mestrado Integrado em Engenharia Informática

Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2018/2019

Implementação da Base de Dados: Livraria Bello

Pedro Machado A33524, Etienne Costa A76089,

Maurício Salgado A71407, Rafael Alves A72629

Janeiro, 2019,

Data de Receção	
Responsável	
Avaliação	
Observações	

Base de Dados: Livraria Bello

**Pedro Machado, Etienne Costa, Maurício
Salgado, Rafael Alves**

Janeiro, 2019

Resumo

Fase 1

O presente relatório visa detalhar todo o processo de elaboração de uma base de dados. Neste caso concreto, a implementação de uma base de dados a ser usada na livraria “Bello”.

Serão registadas todas as etapas da criação de uma base de dados adaptada a um caso concreto de uma atividade comercial. Será efetuado um levantamento de requisitos com o proprietário, que darão origem a um modelo conceptual que vá de encontro às necessidades de gestão da atividade supracitada.

Uma vez validado este modelo, seguir-se-á a sua implementação lógica onde se estruturarão as funcionalidades solicitadas. Após a conclusão do modelo lógico será discutida a implementação física, justificando sempre que necessário, as decisões tomadas, de forma a apresentar a solução proposta face ao problema em questão.

Fase 2

Na fase 2 é descrito todo o processo de migração de um sistema relacional, no caso MySQL, para um sistema não relacional: MongoDB.

Para tal foi construída uma aplicação *Java*, que estabelece a ligação com os dois sistemas, vai buscar os dados à base de dados: MySQL e transfere todo o conteúdo para estruturas, e em seguida a informação será introduzida numa base de dados Mongo.

A estruturação das coleções criadas em Mongo, serão com base nos requisitos de exploração pretendidos para este sistema.

Área de Aplicação: Modelação e Implementação de Sistemas de Bases de Dados

Palavras-Chave: Base de Dados Relacional, Modelação Conceptual, Modelação Lógica, Implementação, MySQL, migração, MongoDB

Índice

RESUMO.....	1
FASE 1.....	1
FASE 2.....	1
ÍNDICE	2
ÍNDICE DE FIGURAS	5
ÍNDICE DE TABELAS	5
1. DEFINIÇÃO DO SISTEMA	6
1.1. CONTEXTO DE APLICAÇÃO DO SISTEMA	6
1.2. FUNDAMENTAÇÃO DA IMPLEMENTAÇÃO DA BASE DE DADOS	6
1.3. ANÁLISE DA VIABILIDADE DO PROCESSO.....	6
2. LEVANTAMENTO E ANÁLISE DE REQUISITOS	7
2.1. MÉTODO DE LEVANTAMENTO E ANÁLISE DE REQUISITOS ADOTADO	7
2.2. REQUISITOS LEVANTADOS.....	8
2.2.1 REQUISITOS DE DESCRIÇÃO	8
2.2.2 REQUISITOS DE EXPLORAÇÃO	8
2.2.3 REQUISITOS DE CONTROLO	8
3. MODELAÇÃO CONCEPTUAL	8
3.1. APRESENTAÇÃO DA ABORDAGEM DE MODELAÇÃO REALIZADA	9
3.2. IDENTIFICAÇÃO E CARACTERIZAÇÃO DAS ENTIDADES	9
3.3. IDENTIFICAÇÃO E CARACTERIZAÇÃO DOS RELACIONAMENTOS	10
3.4. IDENTIFICAÇÃO E CARACTERIZAÇÃO DAS ASSOCIAÇÕES DOS ATRIBUTOS COM AS ENTIDADES E RELACIONAMENTOS	10
ESCRITOR	10
LIVRO	11
CLIENTE	12
FUNCIONÁRIO	12
VENDA	13
RELACIONAMENTO LIVROVENDA	13
3.5. DETALHE OU GENERALIZAÇÃO DE ENTIDADES	13
3.6. DEFINIÇÃO DAS CHAVES PRIMÁRIAS E ALTERNATIVAS	14
3.7. APRESENTAÇÃO E EXPLICAÇÃO DO DIAGRAMA ER	15
3.8. VALIDAÇÃO DO MODELO DE DADOS COM O UTILIZADOR	15
4. MODELAÇÃO LÓGICA.....	16
4.1. CONSTRUÇÃO E VALIDAÇÃO DO MODELO LÓGICO	16
DERIVAR RELACIONAMENTOS A PARTIR DO MODELO CONCEPTUAL.....	16
VALIDAÇÃO ATRAVÉS DA NORMALIZAÇÃO	16
VALIDAÇÃO DAS INTERROGAÇÕES DO UTILIZADOR.....	16
VALIDAÇÃO DAS TRANSAÇÕES ESTABELECIDAS	16

4.2. DESENHO DO MODELO LÓGICO	17
4.3. VALIDAÇÃO DO MODELO ATRAVÉS DA NORMALIZAÇÃO	17
PRIMEIRA FORMA NORMAL	17
SEGUNDA FORMA NORMAL	17
TERCEIRA FORMA NORMAL	17
4.4. VALIDAÇÃO DO MODELO COM INTERROGAÇÕES DO UTILIZADOR	18
4.5. VALIDAÇÃO DO MODELO COM AS TRANSAÇÕES ESTABELECIDAS	19
4.5.1 ADICIONAR UM CLIENTE/FUNCIONÁRIO	19
4.5.2 ADICIONAR UM ESCRITOR	20
4.5.3 ADICIONAR UMA CATEGORIA	20
4.5.4 ADICIONAR UM LIVRO	20
4.5.5 ADICIONAR UMA VENDA	20
4.6. REVISÃO DO MODELO LÓGICO COM O UTILIZADOR	20
5. IMPLEMENTAÇÃO FÍSICA	21
5.1. SELEÇÃO DO SISTEMA DE GESTÃO DE BASES DE DADOS	21
5.2. TRADUÇÃO DO ESQUEMA LÓGICO PARA O SISTEMA DE BASES DE DADOS ESCOLHIDO EM SQL ...	21
5.3. TRADUÇÃO DAS INTERROGAÇÕES ESTABELECIDAS PARA SQL(ALGUNS EXEMPLOS)	22
REQUISITO #1 – PROCURAR LIVRO POR PALAVRA	22
REQUISITO #4 – TOP N LIVROS MAIS VENDIDOS	22
REQUISITO #5 - ESCRITOR COM MAIS VENDAS	22
REQUISITO #8 - TOTAL FATURADO & QUANTIDADE TOTAL DE LIVROS VENDIDOS.	23
REQUISITO #9 - CONJUNTO DE LIVROS QUE NUNCA FORAM ADQUIRIDOS.	23
INTERROGAÇÃO ADICIONAL - FATURAÇÃO NUM INTERVALO DE TEMPO	23
INTERROGAÇÃO ADICIONAL 2 - VENDAS DO LIVRO DO ESCRITOR X COM PARTICIPAÇÃO DO FUNCIONÁRIO Y E O CLIENTE Z	23
5.4. TRADUÇÃO DAS TRANSAÇÕES ESTABELECIDAS PARA SQL(ALGUNS EXEMPLOS)	24
TRANSAÇÃO DA VENDA	24
TRANSAÇÃO DE REGISTO DE CATEGORIA	25
TRANSAÇÃO DE REGISTO DE ESCRITOR	25
TRANSAÇÃO DE REGISTO DE LIVRO	26
TRANSAÇÃO DE REGISTO DE CLIENTE	26
5.5. ESCOLHA, DEFINIÇÃO E CARACTERIZAÇÃO DE ÍNDICES EM SQL (ALGUNS EXEMPLOS)	27
5.6. ESTIMATIVA DO ESPAÇO EM DISCO DA BASE DE DADOS E TAXA DE CRESCIMENTO ANUAL	28
5.6.1 ESTIMATIVA DO ESPAÇO EM DISCO DA BASE DE DADOS	28
5.6.2 TAXA DE CRESCIMENTO ANUAL	29
5.7. DEFINIÇÃO E CARACTERIZAÇÃO DAS VISTAS DE UTILIZAÇÃO EM SQL (ALGUNS EXEMPLOS)	29
LISTAGEM DE TODAS AS VENDAS	29
LISTAGEM DE TODOS OS LIVROS DA LIVRARIA	29
5.8. DEFINIÇÃO E CARACTERIZAÇÃO DOS MECANISMOS DE SEGURANÇA EM SQL (ALGUNS EXEMPLOS)	30
5.9. MECANISMOS DE PRESERVAÇÃO DA INTEGRIDADE DOS DADOS	31

5.10. REVISÃO DO SISTEMA IMPLEMENTADO COM O UTILIZADOR.....	31
6. MIGRAÇÃO PARA SISTEMA NOSQL – MONGO	32
6.1. MOTIVAÇÃO PARA USO DE SISTEMA NOSQL.....	32
6.2. OBJETIVOS DA BASE DE DADOS	32
6.3. REQUISITOS DE EXPLORAÇÃO	33
CLIENTE	33
FUNCIONÁRIO	33
ADMINISTRADOR.....	33
6.4. ESTRUTURAÇÃO DO SISTEMA DE DADOS	34
NO PROCESSO DA MIGRAÇÃO DE DADOS PARA UM SISTEMA DE DADOS MONGODB CONCLUÍMOS QUE, COM BASE NAS	
INTERROGAÇÕES QUE DESEJARÍAMOS FAZER, A NOSSA ESTRUTURA CONTEMPLARIA DUAS COLEÇÕES: LIVROS E VENDAS.	34
LIVROS.....	34
VENDAS	35
6.5. IDENTIFICAÇÃO DE OBJETOS (SQL) ALVO DA MIGRAÇÃO	35
6.6. PROCESSO DE MIGRAÇÃO DE DADOS	36
6.7. MIGRAÇÃO DE DADOS - ETAPAS.....	36
EXTRAÇÃO	36
CONVERSÃO.....	36
CARREGAMENTO.....	36
6.8. IMPLEMENTAÇÃO DO PROCESSO DE MIGRAÇÃO	37
6.9. IMPLEMENTAÇÃO DOS REQUISITOS DE EXPLORAÇÃO	38
#1 PESQUISAR LIVROS DE UM DADO AUTOR.....	38
#2 PESQUISAR LIVROS PELO TÍTULO	38
#3 CONSULTAR STOCK DE UM LIVRO DADO O ISBN	38
#4 VER FATURAÇÃO NUM DADO PERÍODO.....	39
#5 FUNCIONÁRIO COM MAIS VENDAS	39
#6 TOP 5 - CLIENTES COM MAIS COMPRAS.....	40
6.10. REQUISITOS DE CONTROLO.....	41
ADMINISTRADOR – SR. BELLO.....	41
FUNCIONÁRIO	41
CLIENTE	41
7. NOTAS FINAIS	42
FASE 1.....	42
FASE 2.....	42
REFERÊNCIAS	43
ANEXOS.....	44
1. SCRIPT SQL DE CRIAÇÃO	44
2. SCRIPT DE POVOAMENTO.....	47

Índice de Figuras

FIGURA 1 - DIAGRAMA ER	15
FIGURA 2 - MODELO LÓGICO	17
FIGURA 3 - EXEMPLO DE AGREGAÇÃO NO DOCUMENTO LIVRO.....	34
FIGURA 4 EXEMPLO DA ESTRUTURA DE UM DOCUMENTO VENDA.....	35
FIGURA 5 - ESTRUTURAS DE DADOS USADAS NA IMPORTAÇÃO.....	36
FIGURA 6 - CÓDIGO DE FUNÇÃO DE INSERÇÃO DE DADOS	37
FIGURA 7 - INTERROGAÇÃO #1.....	38
FIGURA 8 - OUTPUT INTERROGAÇÃO #1.....	38
FIGURA 9 - INTERROGAÇÃO #2.....	38
FIGURA 10 - OUPUT INTERROGAÇÃO #2	38
FIGURA 11 - INTERROGAÇÃO #3.....	38
FIGURA 12 - OUTPUT INTERROGAÇÃO #3	38
FIGURA 13 - INTERROGAÇÃO #4.....	39
FIGURA 14 - OUTPUT INTERROGAÇÃO #4	39
FIGURA 15 - INTERROGAÇÃO #5.....	39
FIGURA 16 - OUTPUT INTERROGAÇÃO #5	39
FIGURA 17 - INTERROGAÇÃO #6.....	40
FIGURA 18 - OUTPUT INTERROGAÇÃO #6.....	40

Índice de Tabelas

TABELA 1 - ENTIDADES E CARACTERIZAÇÃO	9
TABELA 2 - RELACIONAMENTOS DAS ENTIDADES.....	10
TABELA 3 - ATRIBUTOS DA ENTIDADE ESCRITOR	10
TABELA 4 - ATRIBUTOS DA ENTIDADE LIVRO	11
TABELA 5 - ATRIBUTOS DA ENTIDADE CLIENTE	12
TABELA 6 - ATRIBUTOS DA ENTIDADE FUNCIONÁRIO.....	12
TABELA 7 - ATRIBUTOS DA ENTIDADE VENDA.....	13
TABELA 8 - RELACIONAMENTO ENTRE LIVRO E VENDA.....	13
TABELA 9 - CHAVES DAS ENTIDADES	14
TABELA 10 - CUSTOS DE CADA TABELA	28

1. Definição do Sistema

1.1. Contexto de aplicação do sistema

A livraria Bello, fundada em 1958, está situada na baixa da cidade do Porto, e manteve desde essa data até aos dias de hoje um modelo de negócio tradicional de venda de livros. Trata-se de uma atividade familiar cujo proprietário, Sr. Bello, recentemente adquiriu a totalidade do edifício onde se encontra a loja para assim poder finalmente expandir e modernizar a livraria. Esta modernização e expansão implicará a existência de um sistema informático fiável, visto ser impraticável manter todos os registos em papel, como feito até à data.

1.2. Fundamentação da implementação da base de dados

O aumento das vendas de livros registada nos últimos anos, em parte devido ao afluxo do turismo na cidade, criou a necessidade de uma oferta mais variada de livros em diversas línguas. A expansão pretendida para o negócio acarreta diversas mudanças tanto a nível estrutural como em volume logístico, tornando-se assim indispensável a criação de uma base de dados de forma a controlar o stock existente, bem como a gestão dos funcionários e a clientes.

O aumento da área de loja bem como o consequente aumento de funcionários, torna difícil aferir sobre a existência e localização de um dado livro, tornando imperativo a implementação de um mecanismo que facilite todo o processo de venda de livros sem desvirtuar o conceito tradicional da loja.

1.3. Análise da viabilidade do processo

A expansão pretendida na livraria Bello torna insustentável a continuidade dos registos em papel, o que por si só, mitiga o eventual impacto financeiro do custo de um sistema de base de dados.

Ainda assim, faz sentido analisar o compromisso entre o investimento neste sistema e o eventual proveito trazido pelo mesmo.

A afluência de clientes esperada requer um aumento dos funcionários presentes na loja, e o rápido acesso dos mesmos ao stock existente, bem como um rápido processamento de vendas, implica diretamente em um aumento da eficácia no tempo de atendimento. Consequentemente, o número de funcionários necessários ao bom funcionamento da loja, passará a ser menor, reduzindo assim a despesa com salários.

Os registos das transações de livros permitirão uma análise mais cuidada dos dados existentes, tanto a nível das preferências dos clientes como às categorias mais rentáveis, o que possibilitará diferentes abordagens comerciais de forma a maximizar o lucro.

Registos fidedignos e rapidez de acesso aos mesmos, aumentarão a eficácia e flexibilidade do negócio em relação ao modelo antigo.

Perante estas vantagens, todos os custos relativos à implementação, seja em equipamento seja em software, serão diluídos ao longo do tempo num negócio que terá com melhores condições para prosperar face à realidade competitiva dos dias atuais.

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e análise de requisitos adotado

Após o contacto com o Sr. Bello, seguem-se excertos das trocas de e-mails que se consideraram mais relevantes a nível dos requisitos pretendidos.

Quando inquirido acerca dos dados a armazenar e funcionalidades pretendidas, respondeu:

"Pretendo manter a simplicidade que tem sido a característica do meu negócio, e apenas guardar informações relevantes sem que haja demasiada complexidade."

Quanto aos meus funcionários, pretendo que sejam guardadas as informações básicas como o nome, dados de contato (telefone, e-mail e morada) e identificação (NIF), e uma vez que gostaria de poder controlar os custos, queria que constassem também os vencimentos auferidos pelos mesmos."

"Em relação aos livros quero poder guardar, não só as informações básicas de um livro (ISBN, autor, título, edição, preço, número de páginas, idioma, categoria) como também o número de exemplares em stock."

"(...) As vendas devem igualmente ficar registadas e quero contemplar a possibilidade de descontos ou uso de cupões de bónus assim como saber o funcionário que fez a venda."

"Pretendo que se mantenha o nível de atenção para com os clientes e como tal, gostava de continuar a guardar, para além das informações básicas (nome, telefone, e-mail, morada) a data de nascimento para assim poder mandar mensagens personalizadas em datas festivas ou mesmo cupões."

"(...) Algumas funcionalidades que desejaria ver implementadas seriam:

- *Listagem dos livros mais vendidos;*
- *Autor(es) mais vendido(s);*
- *Vendas discriminadas por funcionário;*
- *Filtrar livros que nunca tenham sido vendidos;*
- *Compras efetuadas por um dado cliente;*
- *Faturação de um dado período.*

"

Seguidamente, inquirimos acerca dos utilizadores da base de dados e o conteúdo que teriam acesso e poderiam alterar:

"Em relação à loja nova, como já foi dito anteriormente, espero conseguir aliviar a carga dos meus empregados e permitir que os clientes façam as pesquisas em terminais espalhados para loja e, como tal, devem poder pesquisar os livros por títulos ou autores e saber se existem exemplares."

"(...) Os funcionários poderão alterar o stock de livros consoante a chegada de novos exemplares de livros e poderão também consultar as vendas da loja."

"(...) Serei o único que poderá consultar e alterar a informação dos funcionários, e obviamente terei acesso a toda a base de dados."

2.2. Requisitos Levantados

2.2.1 Requisitos de descrição

1. Livro: ISBN, título, escritor, editora, edição, ano, páginas, idioma, categoria, stocks, preço;
2. Funcionário: nome, data de nascimento, endereço, telefone, e-mail, vencimento;
3. Cliente: número de cliente, nome, data de nascimento, morada, telefone, e-mail;
4. Venda: livros, quantidades, data, funcionário, valor, desconto efetuado;
5. Escritor: nome, nacionalidade, data de nascimento, biografia.

2.2.2 Requisitos de exploração

1. Procurar livros por título;
2. Procurar livros por Isbn ;
3. Procurar livros por Autor;
4. Lista dos “N” livros mais vendidos;
5. Autor com mais vendas;
6. Pesquisar compras de um dado cliente;
7. Receitas financeiras num dado período;
8. Quantidade de livros vendidos e total faturado;
9. Lista dos livros não vendidos.

2.2.3 Requisitos de controlo

1. Pesquisar livros pode ser efetuado por clientes, funcionários e administrador
2. Atualização de stock e vendas pode ser efetuado por funcionários e administrador
3. Gestão dos funcionários e informação dos mesmos só poderá ser efetuada pelo administrador

3. Modelação Conceptual

Nesta fase, com base no que foi dito pelo proprietário da livraria, tentam-se agrupar os diferentes conceitos, de modo a poder extrair a informação relevante e definir as entidades, relacionamentos e atributos que vão constar no modelo.

3.1. Apresentação da Abordagem de Modelação Realizada

A abordagem usada na modelação conceptual foi:

1. Identificar entidades e relacionamentos;
2. Identificar os atributos das entidades e relacionamentos, e definir o domínio dos mesmos;
3. Determinar as chaves das entidades;
4. Construção do diagrama concetual;
5. Validação com o Utilizador.

3.2. Identificação e caracterização das entidades

Após a análise efetuada aos requisitos segue-se a tabela com as entidades que se consideraram relevantes:

Entidade	Descrição	Informação
Livro	Informação do livro	Isbn; Título; Editora; Edição; Ano; Stock; Páginas; Preço; Idioma; Categoria; Escritor;
Escritor	Informação do autor do livro	Nome; Nacionalidade; Data de Nascimento; Biografia;
Funcionário	Informação do funcionário	Nome; Data de Nascimento; Morada; Contactos; Vencimento;
Cliente	Informação do cliente	Nome; Data de Nascimento; Morada; Contactos;
Venda	Registo de uma transação	Data; Desconto; Total; Funcionário; Cliente;

Tabela 1 - Entidades e Caracterização

3.3. Identificação e caracterização dos relacionamentos

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
Livro	N	Faz Parte	M	Venda
Funcionário	1	Faz	N	Venda
Cliente	1	Participa	N	Venda
Escritor	1	É autor de	N	Livro

Tabela 2 - Relacionamentos das Entidades

3.4. Identificação e caracterização das Associações dos Atributos com as Entidades e Relacionamentos

Escritor

É a entidade que guarda a informação do autor, segue-se a tabela com os seus atributos:

Atributo	Descrição	Tipo de dados	NULL	Multivalor	Derivado
Id_Escritor	Número para identificar escritor	Valor inteiro	X	X	X
Nome	Nome do escritor	Até 45 caracteres	X	X	X
Nacionalidade	Nacionalidade do escritor	Até 45 caracteres	X	X	X
Data de Nascimento	Data de nascimento do escritor	Data	X	X	X
Biografia	Biografia do escritor	Até 300 caracteres	X	X	X

Tabela 3 - Atributos da Entidade Escritor

Livro

Esta entidade armazena a informação de um livro. Segue-se a tabela com os atributos:

Atributo	Descrição	Tipo de dados	NULL	Multivalor	Composto	Derivado
ISBN	ISBN servirá de ID no livro	Valor inteiro	X	X	X	X
Título	Título do livro	Até 75 caracteres	X	X	X	X
Editora	Empresa que publicou o livro	Até 45 caracteres	X	X	X	X
Edição	Número da edição do livro	Valor inteiro	X	X	X	X
Ano	Ano em que o livro foi lançado para o mercado	Valor inteiro	X	X	X	X
Stock	Número de unidades existentes em loja	Valor inteiro	X	X	X	X
Páginas	Número de páginas que o livro possui	Valor inteiro	X	X	X	X
Preço	Preço ao qual o livro é vendido ao público	Valor decimal	X	X	X	X
Idioma	Linguagem em que o livro se encontra escrito	Até 45 caracteres	X	X	X	X
Categoria	Categoria em que se insere o livro (Ex.: Filosofia, Romance, etc)	Valor inteiro	X	X	X	X
Escritor	Pessoa que escreveu o livro	Valor inteiro	X	X	X	X

Tabela 4 - Atributos da Entidade Livro

Cliente

A informação relativa a um cliente da loja:

Atributo	Descrição	Tipo de dados	NULL	Multivalor	Composto	Derivado
Id_cliente	Chave usada para garantir a unicidade do cliente no sistema	Valor inteiro	X	X	X	X
Nome	Nome do cliente	Até 45 caracteres	X	X	X	X
Data de Nascimento	Data de nascimento do cliente	Data	X	X	X	X
Localidade	Morada do cliente (1)	Até 45 caracteres	X	X	✓	X
Rua	Morada do cliente (2)	Até 45 caracteres	X	X	✓	X
Código Postal	Morada do cliente (3)	Até 45 caracteres	X	X	✓	X
Telefone	Número de telefone do cliente	Até 15 caracteres	X	X	X	X
Email	Email de contacto do cliente	Até 45 caracteres	X	X	X	X
(1),(2),(3) - partes integrantes da morada do cliente						

Tabela 5 - Atributos da Entidade Cliente

Funcionário

Esta entidade contém a informação relativa a um funcionário da loja:

Atributo	Descrição	Tipo de dados	NULL	Multivalor	Composto	Derivado
Id_funcionario	Número de identificação do funcionário	Valor inteiro	X	X	X	X
Nome	Nome do funcionário	Até 45 caracteres	X	X	X	X
Data de Nascimento	Data de nascimento do funcionário	Data	X	X	X	X
Localidade	Morada do funcionário (1)	Até 45 caracteres	X	X	✓	X
Rua	Morada do funcionário (2)	Até 45 caracteres	X	X	✓	X
Código Postal	Morada do funcionário (3)	Até 45 caracteres	X	X	✓	X
Telefone	Número de telefone	Até 15 caracteres	X	X	X	X
Vencimento	Salário do funcionário	Float	X	X	X	X
Email	Email do funcionário	Até 45 caracteres	X	X	X	X
(1), (2), (3) - partes integrantes da morada do funcionário						

Tabela 6 - Atributos da Entidade Funcionário

Venda

A entidade 'Venda' guarda os dados relativos a uma venda:

Atributo	Descrição	Tipo de dados	NULL	Multivalor	Composto	Derivado
Id_Venda	Valor inteiro	Chave usada para garantir a unicidade da fatura no sistema	X	X	X	X
Data	Data	Data em que a fatura foi criada	X	X	X	X
Cliente	Valor inteiro	Chave usada para identificação do cliente da fatura	X	X	X	X
Funcionário	Valor inteiro	Chave usada de identificação do funcionário que criou a fatura	X	X	X	X
Cupão	Valor decimal	Desconto percentual aplicado sobre o valor total da fatura	✓	X	X	X
Total	Valor decimal	Valor total da fatura	X	X	X	✓
X - Não; ✓ - Sim						

Tabela 7 - Atributos da Entidade Venda

Relacionamento LivroVenda

Atributo	Tipo de Dados	Descrição	NULL	Multivalor	Composto	Derivado
Preco	Valor decimal	Preço transacionado	X	X	X	X
Quantidade	Valor inteiro	Quantidade comprada do livro	X	X	X	X

Tabela 8 - Relacionamento entre Livro e Venda

3.5. Detalhe ou generalização de Entidades

Não houve necessidade de generalização ou detalhar entidades.

3.6. Definição das Chaves Primárias e Alternativas

Entidade	Chave Primária	Chaves Candidatas	Chave Alternativa
Livro	ISBN (standard internacional)	título	título
Escritor	id_escritor		
Funcionário	id_funcionario	telefone, e-mail	telefone
Cliente	id_cliente	telefone, e-mail	telefone
Venda	id_venda		

Tabela 9 - Chaves das Entidades

3.7. Apresentação e explicação do diagrama ER

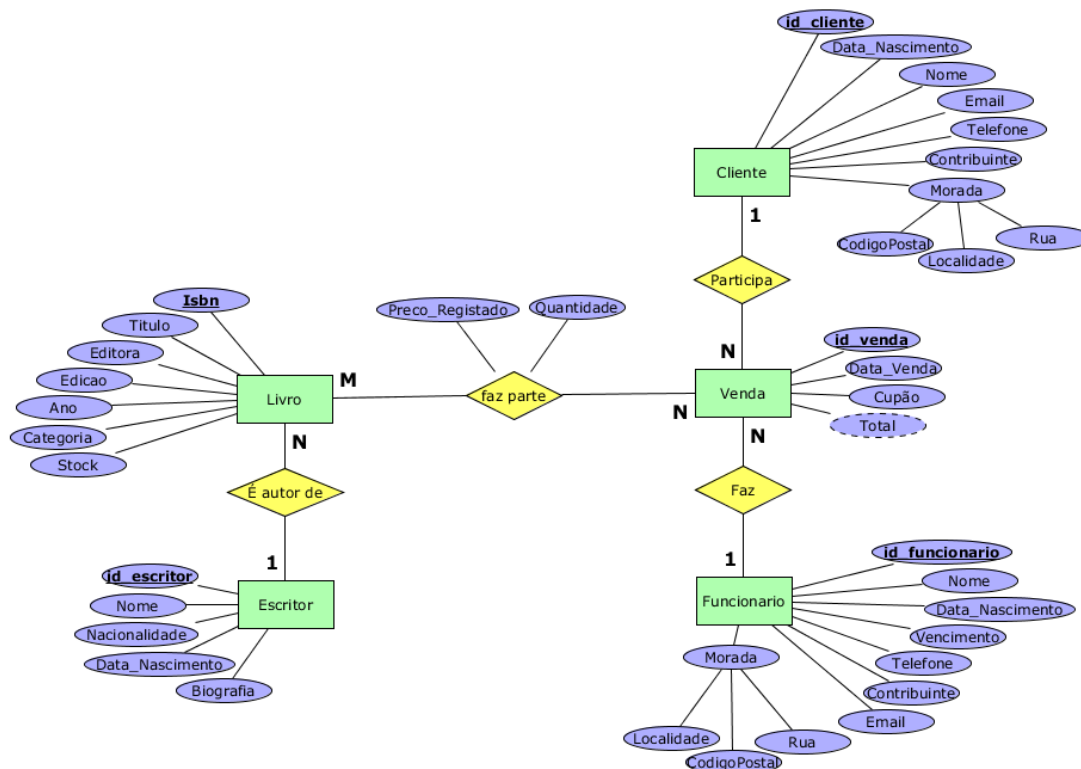


Figura 1 - Diagrama ER

A figura acima usa a notação *Chen* para representar visualmente o modelo conceptual criado, com base nas especificações pedidas.

Como tal estão representadas as 5 entidades: Livro, Escritor, Venda, Funcionário e Cliente

Constam também os atributos de cada entidade sendo que um deles será a chave primária, e estão também representados os relacionamentos entre as várias entidades e a multiplicidade das mesmas.

Podemos também verificar a existência de atributos compostos, no caso a *Morada*, bem como um também atributo derivado na entidade *Venda* que é o *Total* da mesma.

3.8. Validação do modelo de dados com o utilizador

Depois de finalizado o modelo conceptual houve a marcação de um encontro presencial com o proprietário da livraria de forma a discutir o modelo criado.

O Sr. Bello pode constatar que a informação que pretendia guardar está em conformidade com aquilo que tinha solicitado. Após alguma explicação do relacionamento entre as várias entidades, bem como a garantia que poderia efetuar todos os requisitos de exploração e controlo pretendidos, foi possível obter o aval e podemos assim passar para a modelação lógica.

4. Modelação Lógica

Na fase da modelação lógica, é representada a estrutura de dados e a sua organização, abstraindo a implementação física da base de dados. Toda a informação sobre as entidades e relacionamentos entre si estarão presentes neste modelo.

4.1. Construção e Validação do Modelo Lógico

Para a construção deste modelo foram seguidos vários passos:

Derivar relacionamentos a partir do modelo conceptual

- Com base no modelo existente e análise dos relacionamentos, entre *Venda* e *Livro* (N-M) dá origem a uma nova tabela de dados: *LivroVenda*;
- Nos relacionamentos 1-N, a chave primária da entidade com multiplicidade 1 será a chave estrangeira da entidade com multiplicidade N. Exemplo: *Autor* e *Livro*, *Cliente* e *Venda*, *Funcionário* e *Venda*;
- No caso de atributos cujo valor se repita muitas vezes, é boa prática criar uma nova tabela, como foi o caso do atributo *Categoria* em *Livro* que deu origem a um relacionamento 1-N entre *Categoria* e *Livro*;
- O atributo composto *Morada* é decomposto nos seus campos simples.

Validação através da normalização

Nesta fase, descrita posteriormente, validaremos os relacionamentos com base nas formas normais existentes.

Validação das Interrogações do Utilizador

Sendo as interrogações requisitos do utilizador, deverá ser assegurado que o modelo escolhido suporta as mesmas

Validação das Transações Estabelecidas

O modelo terá de suportar as transações requisitadas pelo utilizador

4.2. Desenho do Modelo Lógico

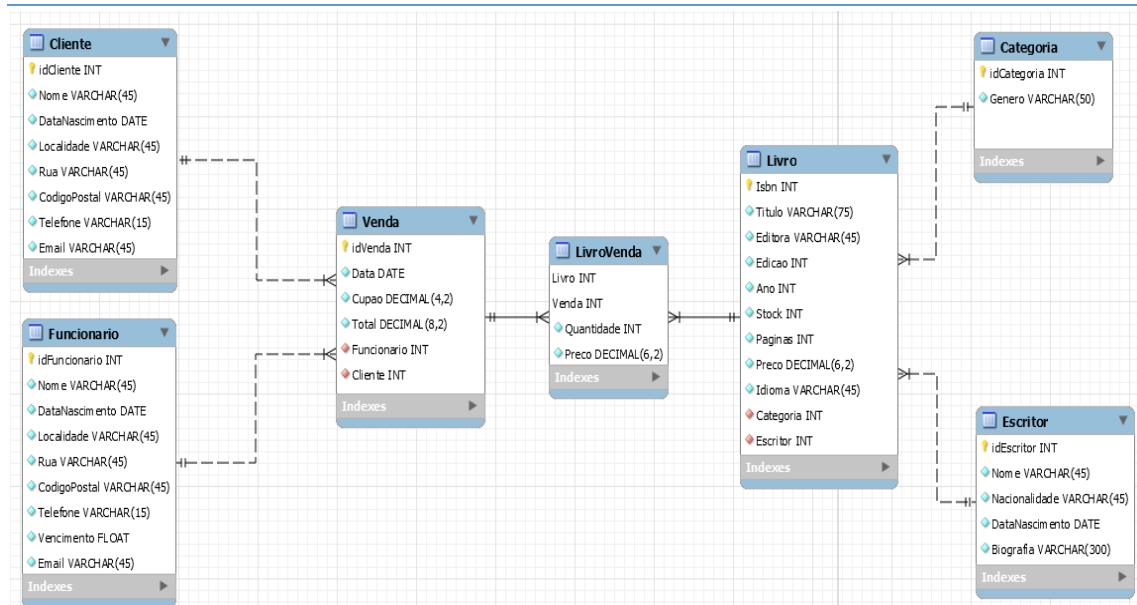


Figura 2 - Modelo Lógico

A figura acima foi o modelo lógico elaborado no software MySQL Workbench.

Podemos constatar a existência de 2 tabelas novas:

- Livro/Venda – surgiu da relação N-M entre as entidades: Livro e Venda
- Categoria – sendo um atributo que se repetia muitas vezes na tabela *Livro*, fez sentido a criação não só de uma nova tabela, como de uma chave estrangeira na entidade *Livro*

4.3. Validação do Modelo através da Normalização

Primeira Forma Normal

Esta forma exige que uma chave primária não tenha correspondência com mais do que um conjunto de atributos. Depois de analisado o modelo lógico podemos constatar que isso nunca aconteceu, portanto podemos afirmar que o mesmo está de acordo com esta forma.

Segunda Forma Normal

A segunda forma normal pressupõe para além da primeira forma normal, que cada atributo que não seja chave esteja dependente da totalidade da chave primária. Após uma análise cuidadosa das relações do nosso modelo, podemos constatar que está de acordo com esta forma.

Terceira Forma Normal

A terceira forma dita que qualquer relação terá de obedecer às duas anteriores formas e, além disso, um atributo (não chave) de uma dada tabela não pode ser calculado através dos restantes atributos, sendo, por isso, independente de todos os restantes. Verificamos que o nosso modelo está de acordo com esta forma.

4.4. Validação do Modelo com Interrogações do Utilizador

Na validação do modelo pretendemos assegurar que o mesmo suporta as interrogações por parte do utilizador, como tal usaremos álgebra relacional para demonstrar a metodologia usada. Asseguraremos assim que será possível efetuar as interrogações requisitadas.

Nota: As interrogações em álgebra relacional poderão não ser uma representação fiel do código implementado das mesmas, servem apenas como meio de validação. No caso de *procedures* serão usados parâmetros exemplo.

1ª Interrogação – Procurar livros por título

$\sigma_{(\text{titulo}='Casa Velha')} \text{Livro}$

2ª Interrogação – Procurar livro por Isbn

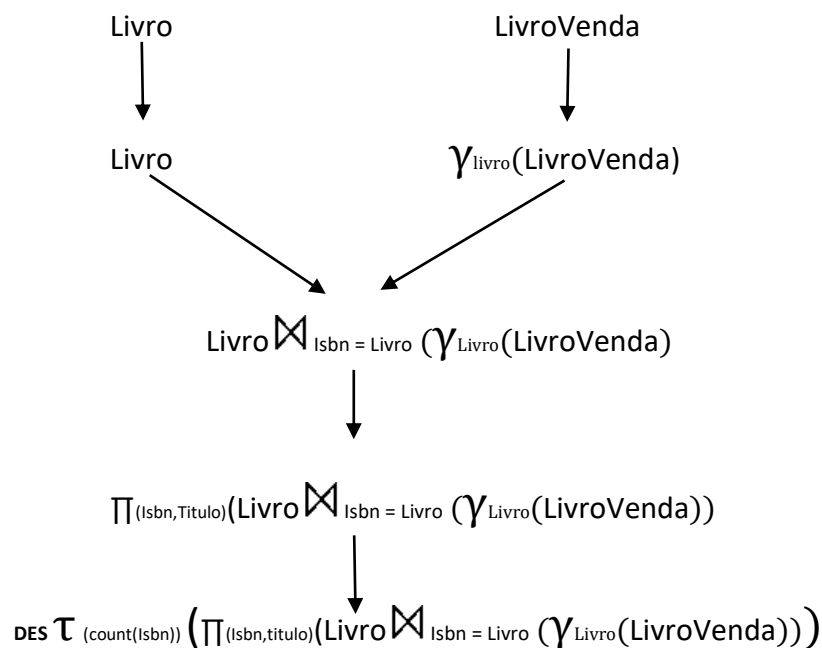
$\sigma_{(\text{Isbn}=12345)} \text{Livro}$

3ª Interrogação – Procurar livros por autor

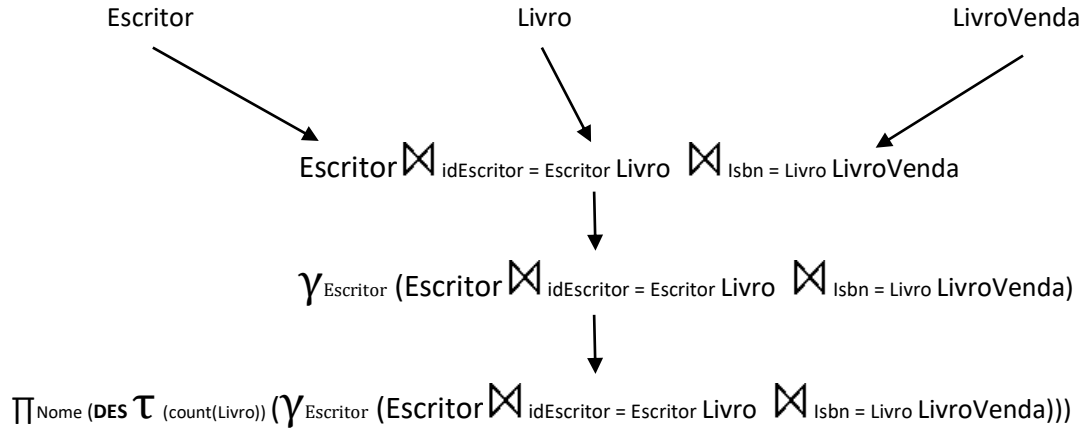
$\text{Livro} \bowtie_{\text{Escritor} = \text{idEscritor}} (\sigma_{(\text{Autor}='jkRowling')} \text{Autor})$

4ª Interrogação – Livros mais vendidos

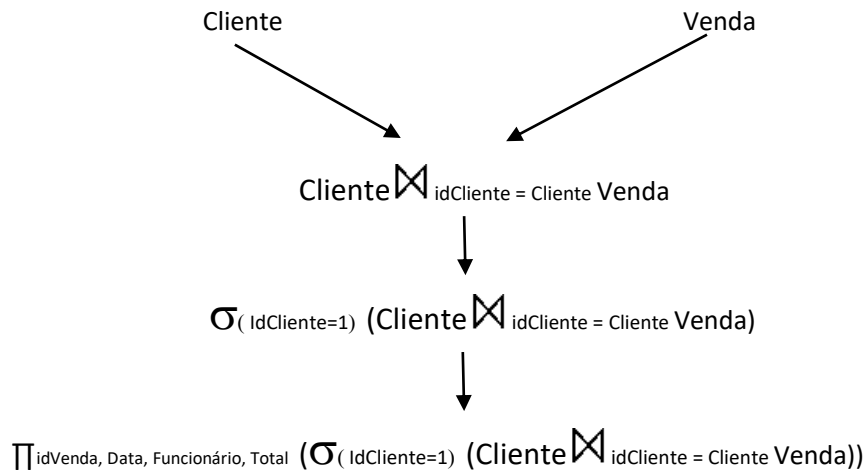
$I4 \leftarrow \tau_{(\text{count}(\text{Isbn}))} (\pi_{(\text{Isbn}, \text{Titulo})} (\text{Livro} \bowtie_{\text{Isbn} = \text{Livro}} (\gamma_{\text{Livro}}(\text{LivroVenda})))$



5ª Interrogação – Autor com mais vendas



6ª Interrogação – Pesquisar compras de um dado cliente



Nas restantes interrogações, devido ao uso de funções não representáveis em álgebra relacional e também à baixa complexidade, decidimos as não representar neste formato.

4.5. Validação do Modelo com as Transações Estabelecidas

As transações ajudam a manter a integridade do sistema em situações em que um determinado procedimento poderá dar erro e gerar inconsistências. Com transações é possível reverter as alterações em caso de erro e manter assim a consistência dos dados.

4.5.1 Adicionar um Cliente/Funcionário

A inserção de um novo cliente/funcionário só é possível caso o e-mail do cliente que se quer registrar ainda não conste do sistema. Aquando da inserção, a chave primária (idCliente/idFuncionario) de Cliente/Funcionário fica presente na tabela Venda como chave estrangeira. Consegue-se, portanto, relacionar Cliente/Funcionário com Vendas.

4.5.2 Adicionar um Escritor

A inserção de um novo Escritor só é possível caso o par Nome e a Data de Nascimento do mesmo ainda não exista no nosso sistema. Aquando da inserção, a chave primária (idEscritor) associa-se à tabela Livro como chave estrangeira. Consegue-se, portanto, relacionar um Escritor aos seus respetivos livros.

4.5.3 Adicionar uma Categoria

A inserção de uma nova Categoria só é possível caso o campo Género ainda não conste da nossa base de dados. A inserção associa uma chave primária (idCategoria) a cada uma das entradas, que é chave estrangeira da tabela Livro. É, portanto, possível associar cada Livro à sua categoria correspondente.

4.5.4 Adicionar um Livro

A inserção de um Livro só é efetuada caso o Isbn do mesmo não exista ainda no sistema. Esta operação só é possível caso o Escritor e a Categoria deste já constem da base de dados.

4.5.5 Adicionar uma Venda

O registo de uma Venda só é possível quando a diferença entre o stock do livro em causa e a quantidade desejada seja maior ou igual a zero. A não verificação desta condição geraria graves inconsistências no sistema.

4.6. Revisão do Modelo Lógico com o Utilizador

Foi combinado mais um encontro com o proprietário da livraria, onde foi discutido o modelo lógico chegado. Mostrou-se o diagrama lógico e o modo como seriam satisfeitos os requisitos pedidos, com transações e interrogações. Foi também explicado como o modelo permitiria registar novos Clientes, Livros e Funcionários e de que forma as vendas, depois de realizadas, atualizariam o stock.

Discutiu-se a forma como o modelo em questão, através de todos os relacionamentos, satisfaria todos os requisitos de exploração pedidos pelo proprietário.

O Sr. Bello confiou na explicação dada e mostrou-se curioso para ver o funcionamento da base de dados.

5. Implementação Física

Nesta secção abordaremos o processo de implementar fisicamente a base de dados a partir do modelo lógico obtido previamente. Será usado o sistema *MySQL* para esta implementação, e podemos encontrar nesta secção a tradução para código *MySQL* de interrogações, transações e criação da base de dados

5.1. Seleção do Sistema de Gestão de Bases de Dados

A escolha do sistema de gestão recaiu para o *MySQL* pois neste caso específico apresenta várias vantagens:

1. É seguro, e neste caso específico foi um dos primeiros pontos chave para o Sr. Bello abdicar do seu tradicional modelo de negócio
2. Custo reduzido quando comparado com as alternativas existentes no mercado
3. Alta Performance no caso de *stored procedures*, que é de extrema importância numa aplicação que terá interrogações constantes usando parâmetros

5.2. Tradução do Esquema Lógico para o Sistema de Bases de Dados escolhido em SQL

Uma vez que foi usado o *MySQL Workbench* na criação do esquema lógico, foi utilizada a ferramenta de *Forward Engineering* na obtenção do esquema físico. O *script* resultante é disponibilizado no **Anexo 1** deste relatório.

5.3. Tradução das Interrogações estabelecidas para SQL(alguns exemplos)

Requisito #1 – Procurar Livro por Palavra

```
DELIMITER $$
CREATE PROCEDURE Searchbyword(IN Chave VARCHAR(45))
BEGIN
SELECT Isbn, Titulo, Editora, Edicao, Ano, Stock, Paginas, Preco, Idioma, C.Genero,
E.Nome as Escritor
FROM Livro
INNER JOIN Escritor AS E ON E.idEscritor = Livro.Escritor
INNER JOIN Categoria AS C ON C.idCategoria = Livro.Categoria
WHERE locate(Chave,Titulo);
ORDER BY isbn;
END
$$
```

Requisito #4 – Top N livros mais vendidos

```
DELIMITER $$
CREATE PROCEDURE TopSellers(IN N INT)
BEGIN
SELECT L.Isbn,L.Titulo,sum(Lv.Quantidade) AS `Quantidade Vendida`,sum(V.Total) AS
`Total Facturado` FROM Livro AS L
INNER JOIN LivroVenda AS Lv ON Lv.Livro = L.Isbn
INNER JOIN Venda AS V ON V.idVenda = Lv.Venda

GROUP BY L.Isbn
ORDER BY `Quantidade Vendida` DESC
LIMIT N;
END
$$
```

Requisito #5 - Escritor com mais vendas

```
DELIMITER $$
CREATE PROCEDURE Topwriters (IN N INT)
BEGIN
SELECT E.* ,count(Lv.Livro) AS `Vendas` FROM Escritor AS E
INNER JOIN Livro AS L ON E.idEscritor=L.Escritor
INNER JOIN LivroVenda AS Lv ON L.Isbn=Lv.Livro
GROUP BY E.idEscritor
ORDER BY `Vendas` DESC
LIMIT N;
END $$
```


Requisito #8 - Total Faturado & Quantidade total de livros vendidos.

```
SELECT * FROM (SELECT SUM(Total) AS `Total Venda` FROM Venda) AS x,  
              (SELECT SUM(Quantidade) AS `Quantidade Vendida` FROM LivroVenda) AS y;
```

Requisito #9 - Conjunto de livros que nunca foram adquiridos.

```
SELECT * FROM Livro  
WHERE Isbn NOT IN (SELECT DISTINCT Livro FROM LivroVenda)  
ORDER BY Isbn;
```

Interrogação adicional - Faturação num intervalo de tempo.

```
DELIMITER $$  
CREATE FUNCTION BillingBetweenDates  
  (inicio date, fim date)  
  RETURNS DECIMAL (6,2) DETERMINISTIC  
BEGIN  
  DECLARE valor DECIMAL (6,2);  
  SELECT SUM(Total) INTO valor FROM Venda  
  WHERE Data BETWEEN inicio AND fim;  
  RETURN valor;  
END $$
```

Interrogação adicional 2 - Vendas do Livro do Escritor X com participação do funcionário Y e o Cliente Z

```
DELIMITER $$  
CREATE PROCEDURE WritterXCustomerXEmployee (IN esc VARCHAR(45), IN cli VARCHAR(45), IN fun  
VARCHAR(45))  
BEGIN  
  SELECT DISTINCT L.Isbn, L.Titulo, C.Genero, E.Nome, Cl.Nome, F.Nome FROM Livro AS L  
  INNER JOIN Escritor AS E ON E.idEscritor=L.Escritor  
  INNER JOIN Categoria AS C ON C.idCategoria=L.Categoria  
  INNER JOIN LivroVenda AS LV ON LV.Livro=L.Isbn  
  INNER JOIN Venda AS V ON V.idVenda=Lv.Venda  
  INNER JOIN Funcionario AS F ON F.idFuncionario=V.Funcionario  
  INNER JOIN Cliente AS Cl ON Cl.idCliente=V.Cliente  
  WHERE (E.Nome=esc AND Cl.Nome=cli AND F.Nome=fun)  
  ORDER BY L.Isbn asc;  
END $$
```

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

Transação da Venda

```
DELIMITER $$
CREATE PROCEDURE compraCliente(IN idCliente INT,IN quantidadex INT,IN idL INT)

BEGIN
    DECLARE idfun INT;
    DECLARE stockx INT;
    DECLARE precox Decimal(6,2);
    DECLARE datav Date;
    DECLARE vId INT;
    DECLARE fail BOOL DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET fail=1;
    set idfun=floor(rand()*(30-1))+1;
    set stockx=(select Stock from Livro where Isbn=idL);
    set precox=(select Preco from Livro where Isbn=idL);
    set datav= date(now());
    set vId=1+ (select Count(idVenda) from Venda ) ;

    START TRANSACTION;
    if(stockx-quantidadex>=0)
    then
    INSERT INTO Venda
        (idVenda,`Data`,Cupao,Total,Funcionario,Cliente)
        VALUES
            (vId,datav,0,0,idfun,idCliente);

    INSERT INTO LivroVenda
        (Livro,Venda,Quantidade, Preco)
        VALUES
            (idL,vId,quantidadex,precox);

    End If;

    If NOT fail Then COMMIT;
    else ROLLBACK;
    END IF;
END
$$
```

Transação de Registo de Categoria

```
DELIMITER $$
CREATE PROCEDURE registaCategoria(IN Genre Varchar(45))
BEGIN
    DECLARE Category INT;
    DECLARE test INT;
    DECLARE fail BOOL DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET fail=1;
    SET Category=1+ (select Count(idCategoria) from Categoria ) ;
    SET test = (select count(Genero) from Categoria
                where Genero=Genre);

    START TRANSACTION;
    if(test=0)
    then
        set fail=1;
    INSERT INTO Categoria
        (idCategoria,Genero)
    VALUES
        (Category,Genre);

    End If;

    If fail Then
    select ('Registo efectuado com sucesso');
    COMMIT;
    else
    select ('Categoria já existe');
    ROLLBACK;
    END IF;
END
$$
```

Transação de Registo de Escritor

```
DELIMITER $$
CREATE PROCEDURE registaEscritor (IN Nomee Varchar(45),IN Nacionalidadee Varchar(45),IN
DataNascimentoo Date,IN Biografiaa Varchar(300))
BEGIN
    DECLARE ID INT;
    DECLARE test INT;
    DECLARE fail BOOL DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET fail=1;
    SET ID =1+ (select Count(idEscritor) from Escritor ) ;
    SET test = (select count(Nome) from Escritor
                where Nome=Nomee and DataNascimento=DataNascimentoo) ;

    START TRANSACTION;
    if(test=0) THEN
        SET fail=1;
    INSERT INTO Escritor
        (idEscritor,Nome,Nacionalidade,DataNascimento,Biografia)
    VALUES
        (ID,Nomee,Nacionalidadee,DataNascimentoo,Biografiaa);

    End If;

    If fail Then
    select ('Registo efectuado com sucesso');
    COMMIT;
    else
    select ('Escritor já existe');
    ROLLBACK;
    END IF;
END
$$
```

Transação de Registo de Livro

```
DELIMITER $$
CREATE PROCEDURE registaLivro (IN Isbnn INT, IN Tituloo Varchar(45), IN Editoraa
Varchar(45), IN Edicaooo INT, IN Anoo INT, IN Stockk INT, IN Paginass INT, IN Precoo decimal
(6,2), IN Idioma Varchar(45), IN Categoriaa Varchar(45), IN Escritorr Varchar(45) )
BEGIN
    DECLARE testIsbn INT;
    DECLARE testE INT;
    DECLARE testC INT;

    DECLARE fail BOOL DEFAULT 0;
    SET testIsbn=(select Isbn from Livro where Isbn=Isbnn);
    SET testE = (select idEscritor from Escritor where Nome= Escritorr );
    SET testC = (select idCategoria from Categoria where Genero= Categoriaa );

    START TRANSACTION;
    if ((testE IS NOT NULL) AND (testC IS NOT NULL) AND testIsbn IS NULL) THEN
        SET fail =1;
    INSERT INTO Livro
        (Isbn,Titulo,Editora,Edicao,Ano,Stock,Paginas,Preco,Idioma,Categoria,Escritor)
        VALUES

        (Isbnn,Tituloo,Editoraa,Edicaooo,Anoo,Stockk,Paginass,Precoo,Idioma,testC,testE);

    End If;
    If fail Then
        select ('Registo efectuado com sucesso');
        COMMIT;
    else
        select ('VERIFICAR SE ESCRITOR,LIVRO ou CATEGORIA JÁ EXISTE!');
        ROLLBACK;
    END IF;
END $$
```

Transação de Registo de Cliente

```
DELIMITER $$
CREATE PROCEDURE registaCliente (IN Nomee Varchar(45), IN DataNascimento Date, IN
Localidadee Varchar(45), IN Ruaa Varchar(45), IN CodigoPostal Varchar(45) , IN Telefonee
Varchar(9), IN Email Varchar(45))
BEGIN
    DECLARE ID INT;
    DECLARE testNome INT;
    DECLARE testEmail INT;
    DECLARE fail BOOL DEFAULT 0;
    SET ID=1+(select Count(idCliente) from Cliente );
    SET testNome = (select Count(idCliente) from Cliente where Nome= Nomee );
    SET testEmail = (select Count(idCliente) from Cliente where Email= Email );

    START TRANSACTION;
    if ((testNome=0) AND (testEmail =0 )) THEN
        SET fail =1;
    INSERT INTO Cliente
        (idCliente, Nome,DataNascimento,Localidade,Rua,CodigoPostal, Telefone,Email)
        VALUES
        (ID,Nomee,DataNascimento,Localidadee,Ruaa,CodigoPostal,Telefonee,Email);

    End If;

    If fail Then
        select ('Registo efectuado com sucesso');
        COMMIT;
    else
        select ('Cliente já está registado!!!');
        ROLLBACK;
    END IF;
END $$
```

5.5. Escolha, Definição e Caracterização de Índices em SQL (alguns exemplos)

Em um contexto mundial onde a quantidade de dados armazenados virtualmente só aumenta, o rápido acesso a eles é de extrema importância, pois o excesso de tempo gasto em uma operação pode acarretar prejuízos. Para combater esse excesso tempo de gasto optamos por implementar um índice sobre a tabela dos livros que nos garante o ganho de performance nas consultas do mesmo.

-- Índice sobre a tabela livros.

```
CREATE INDEX SearchBook ON Livro (Isbn,Titulo);
```

5.6. Estimativa do Espaço em Disco da Base de Dados e Taxa de Crescimento Anual

5.6.1 Estimativa do Espaço em Disco da Base de Dados

Espaço (em bytes) usado por cada entrada na tabela:

	Atributo	Tipo	Espaço Ocupado (bytes)
Livro	Isbn	INT	4
	Titulo	VARCHAR(75)	75
	Editor	VARCHAR(45)	45
	Edição	INT	4
	Ano	INT	4
	Stock	INT	4
	Páginas	INT	4
	Preço	DECIMAL(6,2)	6
	Idioma	VARCHAR(45)	45
	Categoria	INT	4
	Escritor	INT	4
	Total:		199
Escritor	Atributo		Espaço Ocupado (bytes)
	Tipo		
	idEscritor	INT	4
	Nome	VARCHAR(45)	45
	Nacionalidade	VARCHAR(45)	45
	Data de Nascimento	DATE	3
	Biografia	VARCHAR(300)	300
	Total:		397
Cliente	Atributo		Espaço Ocupado (bytes)
	Tipo		
	idCliente	INT	4
	Nome	VARCHAR(45)	45
	Data de Nascimento	DATE	3
	Localidade	VARCHAR(45)	45
	Rua	VARCHAR(45)	45
	Cep	VARCHAR(45)	45
	Telefone	VARCHAR(15)	15
	E-mail	VARCHAR(45)	45
	Total:		247

	Atributo	Tipo	Espaço Ocupado (bytes)
Funcionário	idFuncionario	INT	4
	Nome	VARCHAR(45)	45
	Data de Nascimento	DATE	3
	Localidade	VARCHAR(45)	45
	Rua	VARCHAR(45)	45
	Cep	VARCHAR(45)	45
	Telefone	VARCHAR(15)	15
	Vencimento	FLOAT	4
	E-mail	VARCHAR(45)	45
	Total:		251
LivroVenda	Atributo		Espaço Ocupado (bytes)
	Tipo		
	Livro	INT	4
	Venda	INT	4
	Quantidade	INT	4
Venda	Preço	DECIMAL(6,2)	6
	Total:		18
	Atributo		Espaço Ocupado (bytes)
	Tipo		
	idVenda	INT	4
	Data	DATE	3
	Cupão	DECIMAL(4,2)	4
	Total	DECIMAL(8,2)	8
	Funcionário	INT	4
	Cliente	INT	4
	Total:		27
Categoria	Atributo		Espaço Ocupado (bytes)
	Tipo		
	idCategoria	INT	4
	Genero	VARCHAR(50)	50
	Total:		54

Tabela 10 - Custos de cada tabela

Para conseguir estimar o espaço da base de dados, enviamos um e-mail ao Sr. Bello a solicitar alguns dados sobre expectativas futuras.

Segundo conseguimos apurar com o Sr. Bello, ele conta que o novo espaço tenha:

- 12 empregados
- 10.000 títulos diferentes (de 15 categorias diferentes num universo de 2000 escritores)
- 1300 clientes (valor atual de clientes)
- Atualmente há registos de aproximadamente 6.000 vendas (com média de 2 livros por venda a cada cliente)

Com base nestes dados podemos estimar que o tamanho da base de dados seria:

$$12*251 + 10000*199 + 15*54 + 1300*397 + 6000*27 + 6000*2*18 = 28879220 \text{ bytes} = 27.5 \text{ Mbytes}$$

5.6.2 Taxa de Crescimento Anual

O livreiro referiu também que numa dada semana:

- São editados cerca de 2 livros novos (sendo 1 deles um autor novo)
- Registam-se 3 novos clientes
- Efetuam-se em média 250 vendas (cerca de 2 livros por venda)

Face do exposto, registamos que anualmente a base de dados cresce:

$$52*(2*199 + 3*397 + 250*27 + 250*2*18) = 901628 \text{ bytes} = 0.86 \text{ Mbytes}$$

O que implica um crescimento anual de:

$$0.86/27.5 = 3.1\%$$

5.7. Definição e Caracterização das Vistas de Utilização em SQL (alguns exemplos)

Listagem de todas as vendas

```
CREATE VIEW Faturas AS
SELECT idVenda, F.Nome AS Funcionario, C.Nome AS Cliente, Data, Cupao, Total FROM Venda
  INNER JOIN Funcionario AS F ON F.idFuncionario = Venda.Funcionario
  INNER JOIN Cliente AS C ON C.idCliente = Venda.Cliente
ORDER BY IdVenda
```

Listagem de todos os livros da livraria

```
CREATE VIEW Lista_de_Livros AS
SELECT Isbn, Titulo, Editora, Edicao, Stock, Ano, Paginas, Preco, Idioma, C.Genero, E.Nome AS
Escritor FROM Livro
  INNER JOIN Escritor as E on E.idEscritor=Livro.Escritor
  INNER JOIN Categoria as C on C.idCategoria=Livro.Categoria
ORDER BY Isbn;
```

5.8. Definição e Caracterização dos mecanismos de segurança em SQL (alguns exemplos)

Visto que a nossa base de dados será acedida pelos clientes, funcionários e administrador optou-se por criar perfis de utilizadores para cada um deles, restringindo o acesso de algumas tabelas aos clientes e funcionários visto que nos dias de hoje é essencial a preservação da segurança dos dados.

```
-- Criação do administrador Bello Jr.
CREATE USER 'bellojr'@'localhost';
SET PASSWORD FOR 'bellojr'@'localhost' = 'root';

-- Caracterização geral de um utilizador em particular.
SELECT *
  FROM mysql.user
 WHERE User = 'bellojr';

-- Permissão de acesso a todos os objetos de todas as bases de dados em 'localhost'.
GRANT ALL ON LivrariaBello.* TO 'bellojr'@'localhost';

-- Consulta dos privilégios atribuídos ao utilizador 'bellojr' em 'localhost'.
SHOW GRANTS FOR 'bellojr'@'localhost';

-- Criação do perfil de Funcionário.
CREATE USER 'funcionario'@'localhost';
SET PASSWORD FOR 'funcionario'@'localhost' = 'root';

-- Definição de alguns privilégios para o utilizador 'funcionario'.
-- Permissão para a execução de instruções SELECT, INSERT e UPDATE sobre a base de dados
-- em 'localhost', apenas sobre as tabelas .
GRANT SELECT, INSERT, UPDATE ON LivrariaBello.Livro TO 'funcionario'@'localhost';
GRANT SELECT, INSERT, UPDATE ON LivrariaBello.Escritor TO 'funcionario'@'localhost';
GRANT SELECT, INSERT, UPDATE ON LivrariaBello.Categoria TO 'funcionario'@'localhost';
GRANT SELECT, INSERT, UPDATE ON LivrariaBello.Venda TO 'funcionario'@'localhost';
GRANT SELECT, INSERT, UPDATE ON LivrariaBello.Cliente TO 'funcionario'@'localhost';

-- Permissão para a execução de procedimentos .
GRANT EXECUTE ON PROCEDURE LivrariaBello.* TO 'funcionario'@'localhost';

-- Criação do perfil de Cliente.
CREATE USER 'cliente'@'localhost';
SET PASSWORD FOR 'cliente'@'localhost' = 'root';

-- E apenas de leitura sobre as tabelas Livro e Escritor.
GRANT SELECT ON LivrariaBello.Livro TO 'cliente'@'localhost';
GRANT SELECT ON LivrariaBello.Escritor TO 'cliente'@'localhost';
```


5.9. Mecanismos de Preservação da Integridade dos Dados

De modo a garantir a consistência dos dados, houve a necessidade de implementar dois *triggers*, ambos acionados antes de qualquer inserção na base de dados, que reagem ao mesmo evento - a inserção de dados na tabela *LivroVenda* - que por sua vez faz atualizações na tabela *Venda* (atualizar total) e na tabela *Livro* (atualizar stock) garantindo assim o funcionamento correto da base de dados e integridade dos seus dados.

--AtualizaStock: É responsável pela actualização do stock de um livro na qual uma determinada quantidade foi Vendida.

```
DELIMITER $$
Create Trigger actualizaStock After Insert on LivroVenda
For Each Row
Begin
Update Livro set stock= stock-New.Quantidade
where Isbn=New.Livro;
End;
$$
```

--AtualizaTotal: É responsável pela actualização do total facturado em uma determinada venda.

```
DELIMITER $$
Create Trigger actualizaTotal After Insert on LivroVenda
For Each Row
Begin
Update Venda set Total=Total+ (New.Preco*New.Quantidade) - (New.Preco*New.Quantidade) *Cupao
where idVenda=New.Venda;
End;
$$
```

5.10. Revisão do Sistema Implementado com o Utilizador

Foi marcada uma reunião com o Sr. Bello para demonstrar a funcionalidade (agora visível) da base de dados. O Sr. Bello mostrou-se agradado com os resultados apresentados ao ver satisfeitos todos os requisitos e o funcionamento da base de dados como um todo.

Disse mesmo estar ansioso pela implementação e poder assim prosseguir com a expansão do negócio.

6. Migração para sistema NoSQL – Mongo

6.1. Motivação para uso de sistema NoSQL

Depois do sucesso do sistema prévio de base de dados e a expansão obtida no negócio, o Sr. Bello decidiu que a evolução natural passaria por usar a tecnologia de modo a chegar a novos mercados.

Para comemorar os 113 anos de existência do negócio, decidiu criar um website simples de consulta e encomenda de livros, e após uma reunião com o proprietário, foi explicado que nesta implementação faria sentido optar por um sistema de base de dados relacional. E de seguida foram enumeradas as razões:

- **Escalabilidade:** O MongoDB é escalável horizontalmente, o que ajuda a reduzir a carga de trabalho e escalar o negócio com facilidade, o que iria acontecer com a criação do *website*
- **Velocidade:** Tem performance alta para consultas simples.
- **Dinâmico:** Permite alterar o esquema de dados sem mudar os dados existentes
- **Tolerância a particionamentos:** O sistema continua a funcionar apesar de um número arbitrário de mensagens serem descartadas ou atrasadas pela rede entre nodos.
- **Gestão:** A base de dados não requer administrador, sendo mais *user-friendly* desta forma

Após ouvir atenciosamente estas vantagens e entendendo que seria a escolha acertada para a rápida expansão pretendida, o Sr. Bello ficou estático.

Foi-lhe pedida uma implementação semelhante à anterior a nível de requisitos, e que contemplasse os acessos do site sobre pesquisas de livros. O controlo de permissões iria também manter-se nos mesmos moldes.

6.2. Objetivos da Base de Dados

A rápida expansão do negócio e a implementação de um site, implicam uma quantidade elevada de interrogações acerca de livros e dos seus autores. Então o sistema de base de dados deverá disponibilizar rápido acesso a estes dados sob a forma de interrogações que serão feitas por parte dos utilizadores e funcionários.

Para além disso deverá também suportar algumas interrogações acerca das vendas existentes e os respetivos participantes.

6.3. Requisitos de exploração

Sendo que o sistema será usado por vários tipos de utilizador (clientes, funcionários e administrador), fará sentido implementar as interrogações mais usadas por cada um destes atores.

Cliente

- **#1** - Pesquisar um livro pelo título
- **#2** - Pesquisar livros de um dado autor

O cliente deverá ter acesso apenas de leitura da informação dos livros.

Funcionário

- **#3** - Consultar o stock de um dado livro

Os funcionários devem ter permissões de escrita e leitura para introduzir novos livros, e atualizar o stock dos mesmos.

Administrador

- **#4** - Ver faturação num dado período
- **#5** - Funcionário com mais vendas
- **#6** – Top 5 Cliente(s) com mais compras

Terá permissões de escrita e leitura de todas as coleções de dados, sendo portanto o único com acesso à informação financeira e dados dos seus empregados e clientes.

6.4. Estruturação do Sistema de Dados

No processo da migração de dados para um sistema de dados MongoDB concluímos que, com base nas interrogações que desejaríamos fazer, a nossa estrutura contemplaria duas coleções: **Livros** e **Vendas**.

Livros

Sendo a entidade principal do sistema e também o alvo das interrogações mais usadas, decidiu-se que esta coleção irá ter documentos que agregarão as informações das entidades: **Livro** e o respetivo **Escritor**.

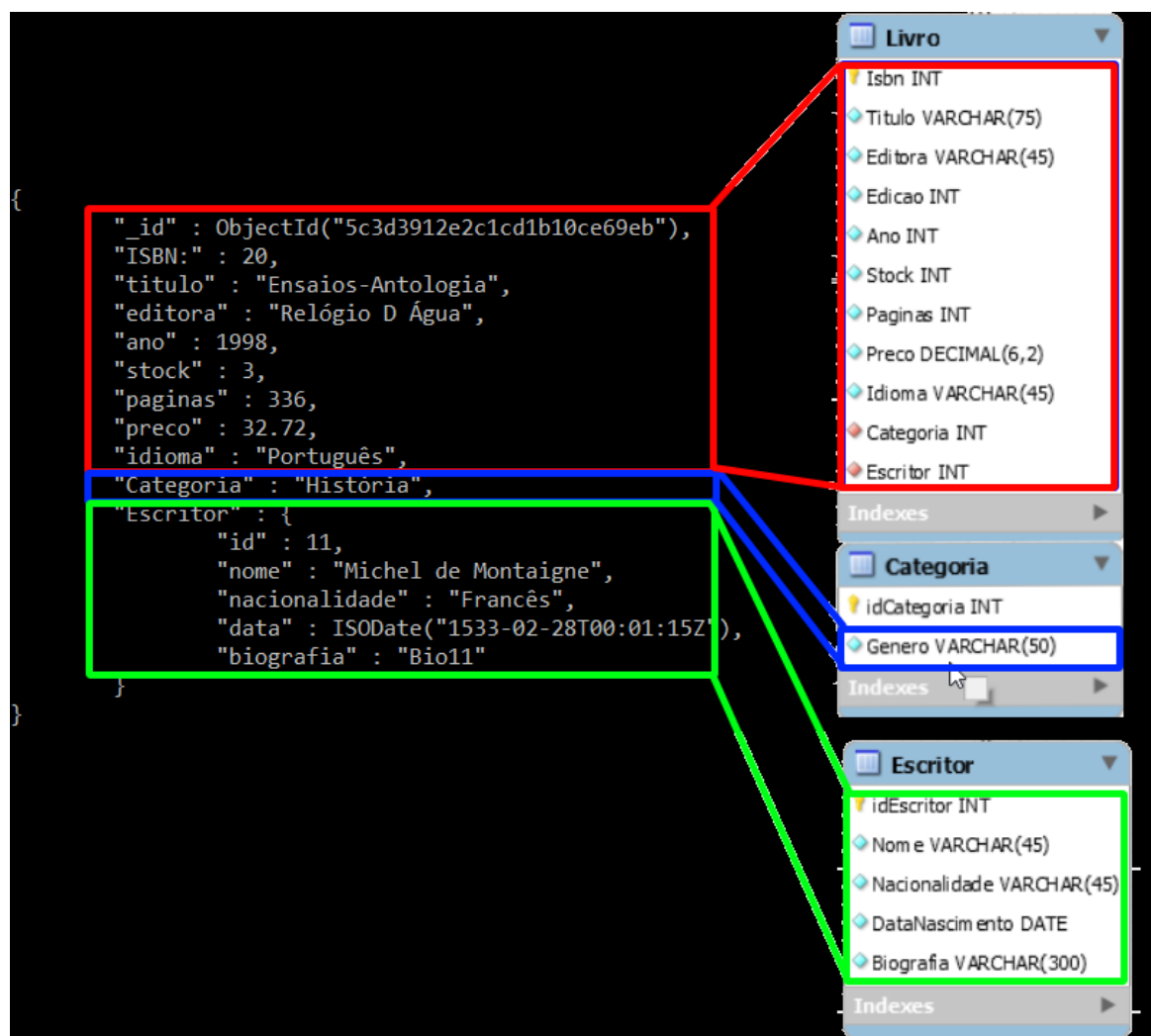


Figura 3 - Exemplo de agregação no Documento Livro

Vendas

Previamente representadas em 4 tabelas (Venda, LivroVenda, Cliente, Funcionario), passa agora a ter documentos com a informação da **Venda**, **Itens** e os dados dos participantes na transação: **Cliente e Funcionário**.

De notar o facto que atributos derivados como o total deixaram de fazer sentido no novo sistema, daí que o na coleção **Venda** o total poderá ser calculando o preço dos diversos itens que contém.

A imagem a seguir exemplifica a estrutura de uma venda:

```
{
  "_id" : ObjectId("5c3d3912e2c1cd1b10ce6a2c"),
  "id" : 20,
  "data" : ISODate("2015-02-15T00:00:00Z"),
  "cupao" : 0,
  "funcionario" : {
    "id" : 7,
    "nome" : "André Maria Gonçalves",
    "datanascimento" : ISODate("1994-07-04T23:00:00Z"),
    "morada" : {
      "localidade" : "Aveiro",
      "rua" : "Rua Maria Felicidade nº130 2ºesq. frente",
      "codigopostal" : "4715-456 Aveiro"
    },
    "telefone" : "917645367",
    "vencimento" : 1000,
    "email" : "amg_1994@gmail.com"
  },
  "cliente" : {
    "id" : 1,
    "nome" : "Anabela Franqueira Alves Rodrigues",
    "data" : ISODate("1963-01-24T00:00:00Z"),
    "morada" : {
      "localidade" : "Matosinhos",
      "rua" : "Avenida Meneses nº234 Bloco 8 1ºesquerdo",
      "codigopostal" : "4715-292 Porto"
    },
    "telefone" : "916275555",
    "email" : "anabela.a.alves@gmail.com"
  },
  "items" : [
    {
      "isbn" : 34,
      "titulo" : "O Homem Mais Feliz da História",
      "quantidade" : 2,
      "preco" : 14.94
    }
  ]
}
```

Figura 4 Exemplo da estrutura de um documento Venda

Em cada venda existe um array de **itens**, em que cada documento corresponde a um livro que integrou uma dada venda. Optou-se por fazer esta distinção entre um documento item e um documento livro, sendo que os documentos livros não estão embebidos na venda, uma vez que dessa forma perderíamos informação da quantidade de cada livro que foi transacionada numa venda. Convém referir que uma venda não necessitará de conter todos estes campos, mas para efeitos de migração decidiu-se incluir os mesmos.

6.5. Identificação de Objetos (SQL) alvo da migração

No caso em concreto, e por forma a ter toda a informação disponível caso seja necessária, todas as tabelas são migradas, sendo que algumas delas não gerarão novas estruturas de dados. Depois de reunida toda a informação, a mesma poderá ser agregada de forma a estruturar da melhor forma o sistema pretendido.

6.6. Processo de migração de dados

Foi criada uma aplicação em Java que estabelece a ligação ao servidor SQL, e guarda a informação das diferentes tabelas em *HashMaps*:

```
private Map<Integer, Livro> livros;  
private Map<Integer, Escritor> escritores;  
private Map<Integer, Funcionario> funcionarios;  
private Map<Integer, Cliente> clientes;  
private Map<Integer, String> categorias;  
private Map<Integer, Venda> vendas;
```

Figura 5 - Estruturas de dados usadas na importação

Houve atenção à ordem como seriam transferidas, pois devido aos diversos relacionamentos as tabelas sem chaves estrangeiras foram transferidas primeiro, para que depois pudessem ser acedidas de forma a completar os dados de outras entidades.

Como dito anteriormente, optou-se por guardar toda a informação e só depois decidir a melhor forma de a usar.

6.7. Migração de Dados - Etapas

Extração

Nesta fase a aplicação estabelece a ligação ao servidor SQL, usando o driver mysql, e vai sequencialmente percorrendo cada tabela e armazenando os seus dados nas estruturas explicitadas anteriormente.

Conversão

Depois de toda a informação estar contida em *HashMaps* cuja chave é o ID dessa entidade, torna-se fácil a manipulação de toda a informação e agregação em Documentos a ser inseridos nas coleções. Esta forma de organização permitirá alterações futuras, como ocultar ou introduzir novos campos de dados com bastante facilidade.

Carregamento

Fazendo uso do driver *mongodb* disponível em Java e de tipos de dados apropriados à gestão deste tipo de base de dados, foi criada uma ligação ao servidor local que permitiu criar coleções e preparar a informação em **Document** que serão criados e carregados na base de dados.

Houve atenção ao formato dos dados, nomeadamente às datas para que respeitassem as normas do MongoDB.

6.8. Implementação do Processo de Migração

Uma vez armazenada toda a informação em estruturas temporárias, HashMaps no caso, as entradas são iteradas e é construído um documento por cada entrada, fazendo os ajustes necessários às datas e também ir buscar a informação que queremos agregar no documento.

A título de exemplo segue a imagem do código:

```
doc = new Document();
doc.put("isbn", l.getIsbn());
doc.put("titulo", l.getTitulo());
doc.put("editora", l.getEditora());
doc.put("ano", l.getAno());
doc.put("stock", l.getStock());
doc.put("paginas", l.getPaginas());
doc.put("preco", l.getPreco());
doc.put("idioma", l.getIdioma());
doc.put("categoria", l.getCategoria());
Escritor e = escritores.get(l.getEscritor());
escritor = new Document();
escritor.put("id", e.getId());
escritor.put("nome", e.getNome());
escritor.put("nacionalidade", e.getNacionalidade());
d = e.getData_nascimento();
instant = d.atStartOfDay().atZone(ZoneId.systemDefault()).toInstant();
try {
    escritor.put("data", Date.from(instant));
}
catch (Exception ex) {
    ex.printStackTrace();
}
escritor.put("biografia", e.getBiografia());
doc.put("Escritor", escritor);
```

Figura 6 - Código de função de inserção de dados

6.9. Implementação dos Requisitos de Exploração

#1 Pesquisar livros de um dado autor

```
db.livros.find(
  {"Escritor.nome": "nome_do_escritor"},
  {_id:0,titulo:1,edicao:1,editora:1,ano:1,preco:1}).pretty()
```

Figura 7 - Interrogação #1

```
{
  "titulo" : "A Confissão da Leoa",
  "editora" : "Editorial Caminho",
  "ano" : 2012,
  "preco" : 12.9
}
{
  "titulo" : "O Beijo da Palavrinha",
  "editora" : "Editorial Caminho",
  "ano" : 2008,
  "preco" : 12.9
}
```

Figura 8 - Output Interrogação #1

#2 Pesquisar livros pelo título

```
db.livros.find(
  {"titulo": "titulo_do_livro"},
  {_id:0,titulo:1,edicao:1,editora:1,ano:1,preco:1}).pretty()
```

Figura 9 - Interrogação #2

```
{
  "titulo" : "Os Maias",
  "editora" : "Livros do Brasil",
  "ano" : 2017,
  "preco" : 11.1
}
```

Figura 10 - Ouput Interrogação #2

#3 Consultar stock de um livro dado o ISBN

```
db.livros.find(
  {"isbn": <isbn>},
  {_id:0,titulo:1,stock:1})
```

Figura 11 - Interrogação #3

```
{ "titulo" : "Mar Me Quer", "stock" : 7 }
```

Figura 12 - Output Interrogação #3

#4 Ver faturação num dado período

```
db.vendas.aggregate([
  {
    "$match": {
      "$and": [
        { data: { $gte: ISODate("2010-04-29T00:00:00.000Z") } },
        { data: { $lte: ISODate("2020-04-29T00:00:00.000Z") } } ]
      }
    },
    {
      "$unwind": "$items"
    },
    {
      "$group":
      {
        _id: null,
        total: { $sum: { $multiply: [ "$items.preco", "$items.quantidade" ] } },
        livrosvendidos: { $sum: "$items.quantidade" }
      }
    },
    {
      $project: { _id: 0, total: 1, livrosvendidos: 1 }
    }
  ]
})
```

Figura 13 - Interrogação #4

```
{ "total" : 7684.66, "livrosvendidos" : 234 }
```

Figura 14 - Output Interrogação #4

#5 Funcionário com mais vendas

```
db.vendas.aggregate(
[
  {
    $group:
    {
      _id: "$funcionario.nome",
      nome: { "$first": "$funcionario.nome" },
      num_vendas: { $sum: 1 }
    },
    {
      $sort: { num_vendas: -1 }
    },
    {
      $limit: 1
    },
    {
      $project: { _id: 0, nome: 1, num_vendas: 1 }
    }
  ]
})
```

Figura 15 - Interrogação #5

```
{ "nome" : "Maria da Conceição Cruz", "num_vendas" : 14 }
```

Figura 16 - Output Interrogação #5

#6 Top 5 - Clientes com mais compras

```
db.vendas.aggregate([
  {
    $group:
    {
      _id: "$cliente.nome",
      nome: { "$first": "$cliente.nome" },
      num_compras: { $sum: 1 },
    }
  },
  {
    $sort: { num_compras: -1 }
  },
  {
    $limit: 5
  },
  {
    $project: { _id: 0, nome: 1, num_compras: 1 }
  }
])
```

Figura 17 - Interrogação #6

```
{ "nome" : "Nuno André Morais Ribeiro", "num_compras" : 12 }
{ "nome" : "Anabela Franqueira Alves Rodrigues", "num_compras" : 10 }
{ "nome" : "Rita Maria Ferreira", "num_compras" : 6 }
{ "nome" : "João Carlos Farelo Marta da Cruz", "num_compras" : 4 }
{ "nome" : "Inês Gonçalves Barros", "num_compras" : 4 }
```

Figura 18 - Output Interrogação #6

6.10. Requisitos de Controlo

Foram criados vários perfis de acordo com as coleções que podem ter acesso

Administrador – Sr. Bello

Deverá ter acesso às vendas e a qualquer coleção

```
db.createUser({
  user: "bello",
  pwd: "bello",
  roles: ["userAdminAnyDatabase", "dbAdminAnyDatabase", "readWriteAnyDatabase"]})
```

Funcionário

Pode aceder à coleção **Livro** com permissões de escrita

```
db.createUser( {
  user: "Funcionario1",
  pwd: "Funcionario1",
  roles: [
    { role: "readWrite", db: "Livro" }
  ]
})
```

Cliente

Apenas pode aceder à coleção **Livro** com permissões de leitura

```
db.createUser( {
  user: "Cliente1",
  pwd: "Cliente1",
  roles: [
    { role: "read", db: "Livro" }
  ]
})
```

7. Notas Finais

Fase 1

Além de aplicar os conhecimentos obtidos nas aulas de Base de Dados redirecionados a uma situação, que se tentou o que fosse o mais realística possível, foi possível entender a necessidade de toda a fase de modelação prévia antes da implementação física de uma base de dados.

A modelação conceptual permitiu que se extraísse unicamente a informação relevante dos requisitos levantados, ajudando assim na eliminação de redundâncias e garantir que a informação relevante era preservada.

A modelação lógica permitiu perceber a melhor forma de estruturar a informação, e como seria de esperar surgiram novas tabelas de dados, resultantes dos relacionamentos identificados no modelo conceptual e também de atributos.

Durante a implementação física foi possível aplicar e aperfeiçoar os conhecimentos da linguagem MySQL e ter noção das potencialidades da mesma.

Entendeu-se que o resultado chegado poderia ainda ser melhorado e tornado mais complexo, e de futuro poder-se-ia fazer uma gestão mais inteligente, por exemplo: uma tabela para gerir armazenar as moradas (uma vez que são um campo comum entre Utilizador e Funcionário), e permitir que alguns atributos como Categoria sejam multivalorados.

Contudo, e com base nos pontos mencionados, achamos que o trabalho desenvolvido permitiu a criação de uma base de dados sólida e íntegra passível de ser implementada num negócio real.

Fase 2

A migração para um sistema NoSQL, neste caso Mongo permitiu descobrir as potencialidades deste tipo de sistema e a diferente abordagem a ser usada num sistema não relacional. Este novo paradigma implica fortes mudanças na estruturação da base de dados, que é facilmente adaptável ao tipo de uso que se procura obter.

As interrogações tornam-se em alguns casos mais simples, pela agregação que é feita num dado documento de múltiplos tipos de dados sem ter que recorrer a relacionamento de tabelas.

A flexibilidade de dados permite que o sistema seja facilmente adaptável, mas mantendo a consistência dos dados existentes, o que permite soluções menos rígidas ao contrário do que acontecia no paradigma relacional que vão de encontro às necessidades de adaptabilidade dos sistemas hoje em dia.

Em relação à migração efetuada, tentou-se que fosse o mais simples e consistente possível e como explicado anteriormente, a aplicação criada em *Java* é facilmente ajustável e flexível a futuras mudanças.

No geral entendeu-se que a solução de migração proposta satisfaz todos os requisitos a nível de consistência dos dados.

Referências

1. Connolly, T.M. and Begg, C.E., 2015. Database Systems, A Pratical Approach to Design, Implementation, and Management. 6th ed. Pearson

Anexos

1. Script SQL de criação

```
-- MySQL Workbench Forward Engineering

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';

-- -----
-- Schema LivrariaBello
-- -----

-- -----
-- Schema LivrariaBello
-- -----

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

-- -----
-- Table `LivrariaBello`.`Cliente`
-- -----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`Cliente` (
  `idCliente` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `DataNascimento` DATE NOT NULL,
  `Localidade` VARCHAR(45) NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `CodigoPostal` VARCHAR(45) NOT NULL,
  `Telefone` VARCHAR(15) NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idCliente`))
ENGINE = InnoDB;

-- -----
-- Table `LivrariaBello`.`Funcionario`
-- -----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`Funcionario` (
  `idFuncionario` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `DataNascimento` DATE NOT NULL,
  `Localidade` VARCHAR(45) NOT NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `CodigoPostal` VARCHAR(45) NOT NULL,
  `Telefone` VARCHAR(15) NOT NULL,
  `Vencimento` FLOAT NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idFuncionario`))
ENGINE = InnoDB;
```

```

-----
-- Table `LivrariaBello`.`Venda`
-----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`Venda` (
  `idVenda` INT NOT NULL,
  `Data` DATE NOT NULL,
  `Cupao` DECIMAL(4,2) NOT NULL,
  `Total` DECIMAL(8,2) NOT NULL,
  `Funcionario` INT NOT NULL,
  `Cliente` INT NOT NULL,
  PRIMARY KEY (`idVenda`),
  INDEX `fk_Venda_Cliente1_idx` (`Cliente` ASC),
  INDEX `fk_Venda_Funcionario1_idx` (`Funcionario` ASC),
  CONSTRAINT `fk_Venda_Cliente1`
    FOREIGN KEY (`Cliente`)
      REFERENCES `LivrariaBello`.`Cliente` (`idCliente`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Venda_Funcionario1`
    FOREIGN KEY (`Funcionario`)
      REFERENCES `LivrariaBello`.`Funcionario` (`idFuncionario`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `LivrariaBello`.`Categoria`
-----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`Categoria` (
  `idCategoria` INT NOT NULL,
  `Genero` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`idCategoria`))
ENGINE = InnoDB;

```

```

-----
-- Table `LivrariaBello`.`Escritor`
-----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`Escritor` (
  `idEscritor` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Nacionalidade` VARCHAR(45) NOT NULL,
  `DataNascimento` DATE NOT NULL,
  `Biografia` VARCHAR(45) NULL,
  PRIMARY KEY (`idEscritor`))
ENGINE = InnoDB;

```

```

-----
-- Table `LivrariaBello`.`Livro`
-----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`Livro` (
  `Isbn` INT NOT NULL,
  `Titulo` VARCHAR(75) NOT NULL,
  `Editora` VARCHAR(45) NOT NULL,
  `Edicao` INT NOT NULL,
  `Ano` INT NOT NULL,
  `Stock` INT NOT NULL,
  `Paginas` INT NOT NULL,
  `Preco` DECIMAL(6,2) NOT NULL,
  `Idioma` VARCHAR(45) NOT NULL,
  `Categoria` INT NOT NULL,
  `Escritor` INT NOT NULL,
  PRIMARY KEY (`Isbn`),
  INDEX `fk_Livro_Categoria1_idx` (`Categoria` ASC),
  INDEX `fk_Livro_Escritor1_idx` (`Escritor` ASC),
  CONSTRAINT `fk_Livro_Categoria1`

```

```

        FOREIGN KEY (`Categoria`)
        REFERENCES `LivrariaBello`.`Categoria` (`idCategoria`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Livro_Escritor1`
        FOREIGN KEY (`Escritor`)
        REFERENCES `LivrariaBello`.`Escritor` (`idEscritor`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `LivrariaBello`.`LivroVenda`
-- -----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`LivroVenda` (
    `Livro` INT NOT NULL,
    `Venda` INT NOT NULL,
    `Quantidade` INT NOT NULL,
    `Preco` DECIMAL(6,2) NOT NULL,
    PRIMARY KEY (`Livro`, `Venda`),
    INDEX `fk_Livro_has_Venda_Venda1_idx` (`Venda` ASC),
    INDEX `fk_Livro_has_Venda_Livro1_idx` (`Livro` ASC),
    CONSTRAINT `fk_Livro_has_Venda_Livro1`
        FOREIGN KEY (`Livro`)
        REFERENCES `LivrariaBello`.`Livro` (`Isbn`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Livro_has_Venda_Venda1`
        FOREIGN KEY (`Venda`)
        REFERENCES `LivrariaBello`.`Venda` (`idVenda`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `LivrariaBello`.`LivroVenda`
-- -----
CREATE TABLE IF NOT EXISTS `LivrariaBello`.`LivroVenda` (
    `Livro` INT NOT NULL,
    `Venda` INT NOT NULL,
    `Quantidade` INT NOT NULL,
    `Preco` DECIMAL(6,2) NOT NULL,
    PRIMARY KEY (`Livro`, `Venda`),
    INDEX `fk_Livro_has_Venda_Venda1_idx` (`Venda` ASC),
    INDEX `fk_Livro_has_Venda_Livro1_idx` (`Livro` ASC),
    CONSTRAINT `fk_Livro_has_Venda_Livro1`
        FOREIGN KEY (`Livro`)
        REFERENCES `LivrariaBello`.`Livro` (`Isbn`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Livro_has_Venda_Venda1`
        FOREIGN KEY (`Venda`)
        REFERENCES `LivrariaBello`.`Venda` (`idVenda`)
        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;

```


2. Script de Povoamento

```
-- Povoamento da tabela "Cliente"
INSERT INTO Cliente
(idCliente, Nome, DataNascimento, Localidade, Rua, CodigoPostal, Telefone, Email)
VALUES
(1, 'Anabela Franqueira Alves Rodrigues', '1963/01/24', 'Matosinhos', 'Avenida
Meneres nº234 Bloco 8 1ºesquerdo',
'4715-292 Porto', '916275555', 'anabela.a.alves@gmail.com'),
(2, 'Marisa Tatiana Morais de Picao', '1994/08/07', 'Vila Nova de Gaia',
'Avenida das Poias nº12 2ºdireito',
'4450-189 Porto', '916473543', 'marisamorais@gmail.com'),
(3, 'Rita Maria Ferreira', '1996/05/18', 'Lisboa',
'Rua do Canudo nº15 1ºdireito',
'3456-133 Lisboa', '911139494', 'ritaferreira@gmail.com'),
(4, 'Luís Filipe Pereira Machado', '1962/01/07', 'Coimbra',
'Praceta João Beltrão nº12 4ºesq. frente',
'3923-456 Coimbra', '935072034', 'lfpm1962@gmail.com'),
(5, 'Daniela Costa Martins Coimbra e Cruz', '1989/04/16', 'Faro',
'Rua da Baixada Porta 8',
'4715-123 Algarve', '911901514', 'danielacruz94@gmail.com'),
(6, 'Inês Gonçalves Barros', '1994/01/26', 'Braga',
'Rua Maria Feliciano nº130 2ºesq. frente',
'4715-234 Braga', '916808958', 'ines_barros@gmail.com'),
(7, 'Mariana Rosário de Matos Araújo', '1994/07/05', 'Aveiro',
'Rua Luís de Camões nº27 5ºesq. trás',
'4715-456 Aveiro', '917645367', 'mariana_araujo@gmail.com'),
(8, 'Tiago Filipe Fernandes Vilaça', '1987/10/04', 'Vila Nova de
Famalicao', 'Rua da Palmeira nº9 1ºesquerdo',
'4715-678 Braga', '917564297', 'tffv_1994@gmail.com'),
(9, 'Pedro Miguel Marinho Almeida', '1993/04/15', 'Leça da Palmeira',
'Rua da Igreja nº13',
'4715-789 Porto', '917283647', 'impiglet@gmail.com'),
(10, 'João Carlos Farelo Marta da Cruz', '1964/01/12', 'São Vitor',
'Rua Maria de Sousa nº4 4ºesq. trás',
'4715-148 Braga', '912749675', 'joaomcruz@gmail.com'),
(11, 'Ana Sofia Franqueira Marta da Cruz', '1996/07/17', 'Matosinhos',
'Rua das Forças Armadas nº8 2ºesq. trás',
'4450-194 Porto', '934678278', 'anasofia_cruz@gmail.com'),
(12, 'Cátia Franqueira Marta da Cruz', '1991/01/12', 'Porto',
'Rua do Pinheiro nº200 2ºesq. frente',
'4450-106 Porto', '934585619', 'catiacruz_1991@gmail.com'),
(13, 'António Gonçalves da Costa Gomes', '1992/08/23', 'Braga',
'Rua São José nº1 2ºesq. trás',
'4450-183 Braga', '936281967', 'ozil_gomes@gmail.com'),
(14, 'Nuno André Morais Ribeiro', '1991/03/29', 'Lisboa',
'Rua Almeida Garret nº15 2ºandar',
'4450-185 Lisboa', '912637486', 'nunoandre91@gmail.com'),
(15, 'Margarida da Silva Ferreira', '1983/06/17', 'Faro',
'Rua do Caixo nº18 7ºfrente',
'4450-190 Faro', '934728917', 'guidaferreira@gmail.com');
```

```

-- Povoamento da tabela "Funcionario"
INSERT INTO Funcionario
(idFuncionario, Nome, DataNascimento, Localidade, Rua, CodigoPostal, Telefone,
Vencimento, Email)
VALUES
(1, 'Maria da Conceição Cruz', '1963/01/24', 'Matosinhos', 'Avenida Meneres
nº234 Bloco 8 1ºesquerdo',
'4715-292 Porto', '916275555', 1000, 'mariacruz@gmail.com'),

(2, 'Beatriz Silva Vilaça', '1994/08/07', 'Vila Nova de Gaia',
'Avenida das Poias nº12 2ºdireito',
'4450-189 Porto', '916473543', 950.50, 'beatrizvilaça@gmail.com'),

(3, 'Beatriz Costa Ferreira', '1996/05/18', 'Lisboa',
'Rua do Canudo nº15 1ºdireito',
'3456-133 Lisboa', '911139494', 700.50, 'beatrizcosta@gmail.com'),

(4, 'Alexandra Sousa de Carvalho', '1962/01/07', 'Coimbra',
'Praceta João Beltrão nº12 4ºesq. frente',
'3923-456 Coimbra', '935072034', 450.50, 'alex_carvalho@gmail.com'),

(5, 'João Miguel Couto', '1989/04/16', 'Faro',
'Rua da Baixada Porta 8',
'4715-123 Algarve', '911901514', 1000, 'joaocouto@gmail.com'),

(6, 'José Pedro Carvalho', '1994/01/26', 'Braga',
'Rua Maria Feliciano nº130 2ºesq. frente',
'4715-234 Braga', '916808958', 450.50, 'zecarvalho@gmail.com'),

(7, 'André Maria Gonçalves', '1994/07/05', 'Aveiro',
'Rua Maria Feliciano nº130 2ºesq. frente',
'4715-456 Aveiro', '917645367', 1000, 'amg_1994@gmail.com'),

(8, 'Leonardo Teixeira Figueiro', '1987/10/04', 'Vila Nova de Famalicão',
'Rua Maria Feliciano nº130 2ºesq. frente',
'4715-678 Braga', '917564297', 950.50, 'leo_figueiro@gmail.com'),

(9, 'Eduardo Nova Silva', '1993/04/15', 'Leça da Palmeira',
'Rua da Igreja nº13',
'4715-789 Porto', '917283647', 700.50, 'eduardosilva@gmail.com'),

(10, 'Maria João Coutinho', '1964/01/12', 'São Vitor',
'Rua Maria de Sousa nº4 5ºesq. frente',
'4715-148 Braga', '912749675', 1000, 'mariacoutinho@gmail.com'),

(11, 'João Eduardo Ferreira', '1996/07/17', 'Matosinhos',
'Rua das Forças Armadas nº8 2ºesq. frente',
'4450-194 Porto', '934678278', 950.50, 'joaoferreira@gmail.com'),

(12, 'André Pereira Marques', '1991/01/12', 'Porto',
'Rua do Pinheiro nº200 2ºesq. frente',
'4450-106 Porto', '934585619', 700.50, 'andremarques_1991@gmail.com'),

(13, 'Rogério Gomes Moreira', '1992/08/23', 'Braga',
'Rua São José nº1 2ºesq. trás',
'4450-183 Braga', '936281967', 450.50, 'rogerio_gomes@gmail.com'),

(14, 'Samuel Gonçalves Ferreira', '1991/03/29', 'Lisboa',
'Rua Almeida Garret nº15 2ºandar',
'4450-185 Lisboa', '912637486', 950.50, 'samu_ferreira@gmail.com'),

(15, 'Tiago Filipe Sá', '1983/06/17', 'Faro',
'Rua do Caixo nº18 7ºfrente',
'4450-190 Faro', '934728917', 700.50, 'tiagofilipesa@gmail.com'),

(16, 'Ricardo Manuel de Sousa e Pinto', '1963/01/24', 'Matosinhos',
'Avenida Meneres nº234 Bloco 8 1ºesquerdo',
'4715-292 Porto', '916275555', 1000, 'ricardopinto@gmail.com'),

(17, 'Pedro Miguel Barroso', '1994/08/07', 'Vila Nova de Gaia',
'Avenida das Poias nº12 2ºdireito',
'4450-189 Porto', '916473543', 700.50, 'pedrobarroso@gmail.com'),

(18, 'Nuno Rosário dos Reis', '1996/05/18', 'Lisboa',
'Rua do Canudo nº15 1ºdireito',
'3456-133 Lisboa', '911139494', 950.50, 'nuno_reis@gmail.com'),

(19, 'Liliana de Castro Pereira', '1962/01/07', 'Coimbra',
'Praceta João Beltrão nº12 4ºesq. frente',
'3923-456 Coimbra', '935072034', 450.50, 'lilianapereira1@gmail.com'),

(20, 'Nuno Bento Armada', '1989/04/16', 'Faro',
'Rua da Baixada Porta 8',
'4715-123 Algarve', '911901514', 1000, 'nunoarmada89@gmail.com'),

(21, 'Mariana Fernandes de Brito', '1994/01/26', 'Braga',
'Rua Maria Feliciano nº130 2ºesq. frente',
'4715-234 Braga', '916808958', 950.50, 'marianabrito94@gmail.com'),

(22, 'Marcelo Filipe Lima', '1994/07/05', 'Aveiro',
'Rua Maria Feliciano nº130 2ºesq. frente',
'4715-456 Aveiro', '917645367', 700.50, 'marcelolima94@gmail.com'),

```

```

(23, 'Paulo Castro de Freitas', '1987/10/04', 'Vila Nova de Famalicão',
'Rua Maria Feliciano nº130 2ºesq. frente',
'4715-678 Braga', '917564297', 450.50, 'paulofreitas@gmail.com'),
(24, 'Luis Leite Ferraz', '1993/04/15', 'Leça da Palmeira',
'Rua da Igreja nº13',
'4715-789 Porto', '917283647', 950.50, 'luisferraz@gmail.com'),
(25, 'José Gomes da Cunha', '1964/01/12', 'São Vitor',
'Rua Maria de Sousa nº4 5ºesq. frente',
'4715-148 Braga', '912749675', 700.50, 'josedacunha@gmail.com'),
(26, 'João Miguel Amorim Fernandes', '1996/07/17', 'Matosinhos',
'Rua das Forças Armadas nº8 2ºesq. frente',
'4450-194 Porto', '934678278', 950.50, 'joaoamorim@gmail.com'),
(27, 'Joana Maria Feijó', '1991/01/12', 'Porto',
'Rua do Pinheiro nº200 2ºesq. frente',
'4450-106 Porto', '934585619', 1000, 'joanafeijo@gmail.com'),
(28, 'Inês Guerreiro Vasconcelos', '1992/08/23', 'Braga',
'Rua São José nº1 2ºesq. trás',
'4450-183 Braga', '936281967', 950.50, 'inesvasconcelos@gmail.com'),
(29, 'Fábio Vieira Mendes', '1991/03/29', 'Lisboa',
'Rua Almeida Garret nº15 2ºandar',
'4450-185 Lisboa', '912637486', 450.50, 'fabiomendes91@gmail.com'),
(30, 'Bruna Carvalho Cardoso', '1983/06/17', 'Faro',
'Rua do Caixo nº18 7ºfrente',
'4450-190 Faro', '934728917', 1000, 'brunacardoso@gmail.com');

```

-- Povoamento da tabela "Escrivor"

```

INSERT INTO Escritor
(idEscrivor, Nome, Nacionalidade, DataNascimento, Biografia)
VALUES
(1, 'Jorge Amado', 'Brasileiro', '1912/08/10', 'Bio1'),
(2, 'Paulo Coelho', 'Brasileiro', '1947/08/24', 'Bio2'),
(3, 'José Saramago', 'Português', '1922/11/16', 'Bio3'),
(4, 'Eça de Queiroz', 'Português', '1845/11/25', 'Bio4'),
(5, 'Fiódor Dostoiévski', 'Russo', '1821/10/30', 'Bio5'),
(6, 'Augusto Cury', 'Brasileiro', '1958/10/02', 'Bio6'),
(7, 'José Bernado Barros', 'Português', '1965/05/22', 'Bio7'),
(8, 'Henry Miller', 'Norte Americano', '1891/12/26', 'Bio8'),
(9, 'Charles Dickens', 'Britânico', '1812/02/07', 'Bio9'),
(10, 'Machado de Assis', 'Brasileiro', '1839/06/21', 'Bio10'),
(11, 'Michel de Montaigne', 'Francês', '1533/02/28', 'Bio11'),
(12, 'Joanne Murray', 'Britânica', '1965/07/31', 'Bio12'),
(13, 'Stephen King', 'Norte Americano', '1947/09/21', 'Bio13'),
(14, 'Pepetela', 'Angolano', '1941/10/29', 'Bio14'),
(15, 'Mia Couto', 'Moçambicano', '1955/07/05', 'Bio15');

```

-- Povoamento da tabela "Categoria"

```

INSERT INTO Categoria
(idCategoria, Genero)
VALUES
(1, 'Romance'),
(2, 'Gastronomia'),
(3, 'Ficção'),
(4, 'Filosofia'),
(5, 'História'),
(6, 'Guerra'),
(7, 'Medicina'),
(8, 'Música'),
(9, 'Psicologia'),
(10, 'Terror'),
(11, 'Contos'),
(12, 'Policial'),
(13, 'Teatro'),
(14, 'Quadrinhos'),
(15, 'Thriller'),
(16, 'Drama'),
(17, 'Educação');

```

```
-- Povoamento da tabela "Livro"
INSERT INTO Livro
(Isbn,Titulo,Editora,Edicao,Ano,Stock,Paginas,Preco,Idioma,Categoria,Escritor)
VALUES
(1,'Sua Excelência de Corpo Presente','Dom
Quixote',9,2018,15,272,15.21,'Português',1,14),
(2,'Parábola do Cágado velho','Dom
Quixote',12,2006,13,160,13.90,'Português',1,14),
(3,'Se o Passado Não Tivesse Asas','Dom
Quixote',5,2016,10,384,18.90,'Português',1,14),
(4,'A Montanha da Água Lilas','Bis',1,2009,20,160,5.95,'Português',1,14),
(5,'A Confissão da Leoa','Editorial Caminho',4,2012,25,272,12.90
,'Português',1,15),
(6,'O Beijo da Palavrinha','Editorial Caminho',11,2008,30,32,12.90
,'Português',3,15),
(7,'Mar Me Quer','Editorial Caminho',9,2012,7,72,9.90
,'Português',1,15),
(8,'A Varanda do Frangipani','Editorial Caminho',4,2000,10,154,12.90
,'Português',1,15),
(9,'IT','HODDER & STOUGHTON GENERAL DIVISION',5,2011,45,1184,13.28
,'Inglês',15,13),
(10,'The Shining','11X17',1,2017,33,632,11.00,'Inglês',15,13),
(11,'Elevation','HODDER & STOUGHTON GENERAL DIVISION',10,2018,20,160,
16.30,'Inglês',15,13),
(12,'Sr.Mercedes','Bertrand Editora',3,2017,5,472,18.80
,'Português',15,13),
(13,'Fantastic Beasts: The Crimes Of Grindelwald','Little,Brown Book Group',
11,2018,13,304,15.96,'Inglês',11,12),
(14,'Harry Potter e a Pedra Filosofal','Editorial Presença',4,2002
,10,260,15.90,'Português',11,12),
(15,'Uma Morte Súbita','Editorial Presença',11,2012,15,496,22.90
,'Português',1,12),
(16,'Fantastic Beasts And Where To Find Them','Little,Brown Book Group',7,
2018,5,304,8.69,'Inglês',12,12),
(17,'Dictionnaire DE Michek De Montaigne','Classiques Garnier',11,2018
,15,2000,56.10,'Francês',9,11),
(18,'Essais','Gallimard',9,2009,20,10,27.91,'Francês',1,11),
(19,'Vivre à Propos','Flammarion',3,2009,7,256,24.17,'Francês',4,11),
(20,'Ensaaios-Antologia','Relógio D Água',4,1998,3,336,32.72
,'Português',5,11),
(21,'O Alienista','Porto Editora',10,2017,20,112,8.80
,'Português',11,10),
(22,'Dom Casmurro','BIS',3,2013,10,320,7.50,'Português',1,10),
(23,'Casa Velha','Narrativa',6,2018,5,128,10.80,'Português',1,10),
(24,'Memórias Póstumas de Brás Cubas','Relógio D Água',02,2017,15,232,
10.00,'Português',16,10),
(25,'Oliver Twist','Engage Books',11,2018,20,400,48.29,'Inglês',3,9),
(26,'The Old Curiosity Shop','Engage Books',11,2018,9,508,54.32
,'Inglês',3,9),
(27,'A Tale of Two Cities','Engage Books',11,2018,11,344,54.32
,'Inglês',3,9),
(28,'Hard Times','Engage Books',11,2018,9,252,54.32,'Inglês',3,9),
(29,'Essays On Fielding Miscellanies','Princeton University Press',8,2018
,24,490,142.47,'Inglês',5,8),
(30,'Sexus','PENGUIN BOOKS LTD',8,2015,5,512,12.07,'Inglês',3,8),
(31,'Plexus','PENGUIN BOOKS LTD',8,2015,9,528,12.07,'Inglês',3,8),
(32,'Tropic Of Cancer','PENGUIN BOOKS LTD',6,2015,9,272,12.07
,'Inglês',3,8),
(33,'Fundamentos da Computação :Programação Funcional','PENGUIN BOOKS LTD',2
,2015,32,100,25.00,'Português',17,7),
(34,'O Homem Mais Feliz da História','Pergaminho',10,2018,15,336,14.94
,'Português',1,6),
(35,'Pais Brilhantes,Professores Fascinantes','Pergaminho',4,2004,14,176,
13.30,'Português',17,6),
(36,'Crime And Punishment','PENGUIN BOOKS LTD',1,2003,9,720,9.66
,'Inglês',3,5),
(37,'O Jogador','Editorial Presença',4,2001,3,168,12.90,'Inglês',1,5),
(38,'Os Maias','Livros do Brasil',8,2017,50,736,11.10,'Português',1,4),
(39,'Memorial do Convento','Porto Editora',1,2018,30,402,17.70
,'Português',1,3),
(40,'Ensaio sobre a Cegueira','Porto Editora',7,2018,10,272,15.93
,'Português',1,3),
(41,'O Alquimista','11X17',10,2013,12,200,8,'Português',1,2),
(42,'A Espia','Pergaminho',9,2016,2,184,15.50,'Português',1,2),
(43,'Capitães da Areia','BIS',1,2009,6,208,7.50,'Português',1,1),
```

```
(44, 'Gabriela,Cravo e Canela', 'BIS' , 10 , 2018 ,12,464, 8.96
,'Português',1,1);
```

```
-- Povoamento da tabela "Venda"
```

```
INSERT INTO Venda
(idVenda, `Data`,Cupao,Total,Funcionario,Cliente)
VALUES
(1, '2011/03/22',0,0,1, 3),
(2, '2011/03/22',0.25,0, 1, 12),
(3, '2011/03/22',0,0, 1, 10),
(4, '2011/03/22',0.1,0, 2, 14),
(5, '2011/05/10',0,0, 2, 1),
(6, '2011/05/11',0,0, 2, 13),
(7, '2011/05/11',0,0, 2, 2),
(8, '2011/05/22',0,0, 3, 8),
(9, '2011/11/11',0.15,0, 3, 10),
(10, '2011/12/01',0,0, 9, 15),
(11, '2013/01/02',0,0, 30, 11),
(12, '2013/03/22',0,0, 30, 5),
(13, '2013/03/22',0,0, 30, 3),
(14, '2013/03/22',0.30,0, 30, 14),
(15, '2014/09/08',0,0, 23, 7),
(16, '2014/11/13',0.10,0, 26, 14),
(17, '2014/11/17',0,0, 17, 1),
(18, '2015/01/19',0,0, 27, 1),
(19, '2015/02/14',0.15,0, 1, 3),
(20, '2015/02/15',0,0, 7, 1),
(21, '2015/06/12',0,0, 19, 15),
(22, '2015/06/12',0.05,0, 19, 14),
(23, '2016/07/06',0.10,0, 1, 6),
(24, '2016/07/17',0,0, 1, 6),
(25, '2016/08/16',0,0, 1, 14),
(26, '2016/10/09',0,0, 30, 1),
(27, '2017/04/03',0.20,0, 30, 9),
(28, '2017/04/22',0,0, 20, 4),
(29, '2018/08/22',0,0, 21, 14),
(30, '2018/08/22',0,0, 21, 2);
```

```

-- Povoamento da tabela "LivroVenda"
INSERT INTO LivroVenda
(Livro,Venda,Quantidade, Preco)
VALUES
(1,1,3, 15.21),
(2, 1,2, 13.90),
(1, 2,1, 15.21),
(2, 2,3, 13.90),
(44,2,2, 8.96 ),
(7, 3,2, 9.90),
(8, 4,1, 12.90),
(9, 4,4, 13.28),
(9, 5,4, 13.28),
(11,6,1, 16.30),
(23,6,1, 10.80),
(23,7,2, 10.80),
(27,8,3, 54.32),
(28,8,1, 54.32),
(29,8,13,142.47),
(29,9,1, 142.47),
(29,10,1,142.47),
(33,11,1,25.00),
(33,12,5,25.00),
(39,12,2,17.70),
(40,13,4,15.93),
(41,14,6,8),
(42,14,1,15.50),
(44,15,3,8.96 ),
(1,16,4, 15.21),
(19,16,5,24.17),
(1,17,1, 15.21),
(1,18,1, 15.21),
(1,19,1, 15.21),
(23,19,1,10.80),
(34,20,2, 14.94),
(36,21,7, 9.66),
(10,22,6,11.00),
(24,23,4, 10.00),
(16,24,4, 8.69),
(35,25,1, 13.30),
(26,26,4, 54.32),
(44,27,5, 8.96 ),
(1,28,1, 15.21),
(1,29,1, 15.21),
(1,30,2, 15.21);

```