



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2020/2021

Desenvolvimento de uma plataforma de venda online de produtos desportivos

**António Silva (89558), Diogo Araújo (89517),
Mário Coelho (42865), Pedro Pereira (80376)**

Dezembro, 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Desenvolvimento de uma plataforma de venda online de produtos desportivos

**António Silva (89558), Diogo Araújo (89517),
Mário Coelho (42865), Pedro Pereira (80376)**

Dezembro, 2020

Resumo

O presente relatório resume os principais trabalhos realizados no âmbito do desenvolvimento de uma plataforma de venda online de produtos desportivos. Mais concretamente, serão detalhadas as tarefas relativas ao sistema de base de dados que dará suporte à referida plataforma.

A metodologia adotada no presente trabalho foi a preconizada por Connolly e Begg (2004). Foram utilizadas algumas ferramentas comuns neste tipo de trabalho, em particular, foi adotado o modelo relacional de bases de dados com recurso ao mysql.

As condicionantes da plataforma, em termos da sua utilização futura, permitiram estabelecer um conjunto de requisitos que se procurou respeitar de forma eficiente. Para tal foi crucial uma estreita comunicação com o cliente ao longo de todo o processo de desenvolvimento.

O objetivo inicialmente definido foi considerado atingido. Com efeito, todos os requisitos que o cliente tinha solicitado foram implementados na base de dados. Adicionalmente, o sistema foi concebido de modo a que facilmente cresça no futuro, através da inclusão de novos de novas categorias e tabelas.

Área de Aplicação: Desenvolvimento de base de dados relacional

Palavras-Chave: Produtos desportivos, e-commerce, sistema de gestão, base de dados relacional.

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	v
1. Definição do Sistema	1
1.1. Contexto de aplicação do sistema	1
1.2. Fundamentação da implementação da base de dados	2
1.3. Análise da viabilidade do processo	2
1.4. Estrutura do Relatório	2
2. Levantamento e Análise de Requisitos	4
2.1. Método de levantamento e de análise de requisitos adotado	4
2.2. Requisitos levantados	4
2.3. Análise e validação geral dos requisitos	6
2.3.1 Requisitos de Descrição	6
2.3.2 Requisitos de Exploração	7
2.3.3 Requisitos de Controlo	8
3. Modelação Conceptual	9
3.1. Identificação e caracterização das entidades	9
3.2. Identificação e caracterização dos relacionamentos	11
3.3. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	12
3.4. Detalhe ou generalização de entidades	16
3.5. Apresentação e explicação do diagrama ER	16
3.6. Validação do modelo de dados produzido	17
4. Modelação Lógica	21
4.1. Construção e validação do modelo de dados lógico	21
4.2. Desenho do modelo lógico	22
4.3. Validação do modelo através da normalização	23
4.4. Validação do modelo com interrogações do utilizador	24

4.5. Reavaliação do modelo lógico (se necessário)	24
4.6. Revisão do modelo lógico produzido	25
5. Implementação Física	26
5.1. Seleção do sistema de gestão de bases de dados	26
5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	26
5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)	27
5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)	29
5.5. Estimativa do espaço em disco da base de dados e taxa de crescimento anual	29
5.6. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	31
5.7. Revisão do sistema implementado	31
6. Conclusões e Trabalho Futuro	32
Referências Bibliográficas	34
Lista de Siglas e Acrónimos	35
Anexos	36
I. Anexo 1 – Script para geração do modelo físico	37
II. Anexo 2 – Script para povoamento do modelo físico	44
III. Anexo 3 – Script para execução de queries	50

Índice de Figuras

Figura 1 – Diagrama ER do sistema.	16
Figura 2 – A entidade Cliente guarda todas as informações referidas.	17
Figura 3 – Diagrama ER do sistema (zoom no relacionamento compra-produto).	18
Figura 4 – Diagrama ER do sistema (zoom no relacionamento loja-produto).	18
Figura 5 – Diagrama ER do sistema (zoom no relacionamento compra-produto).	19
Figura 6 – Diagrama ER do sistema (zoom no relacionamento cliente-compra).	19
Figura 7 – Diagrama ER do sistema (zoom).	20
Figura 8 – Modelo lógico do sistema.	23

Índice de Tabelas

Tabela 1 – Relacionamentos existentes na base de dados.	11
Tabela 2 – Associação dos atributos com a entidade Produto.	12
Tabela 3 – Associação dos atributos com a entidade Cliente.	13
Tabela 4 – Associação dos atributos com a entidade Fornecedor.	13
Tabela 5 – Associação dos atributos com a entidade Funcionário.	13
Tabela 6 – Associação dos atributos com a entidade Compra.	14
Tabela 7 – Associação dos atributos com a entidade Loja.	14
Tabela 8 – Associação dos atributos com a entidade Armazém.	14
Tabela 9 – Associação dos atributos com o relacionamento Contem (Produto-Compra).	15
Tabela 10 – Associação dos atributos com o relacionamento Fornecido (Produto-Fornecedor).	15
Tabela 11 – Associação dos atributos com o relacionamento Tem (Produto-Armazém).	15
Tabela 12 – Associação dos atributos com o relacionamento Tem (Produto-Loja).	15
Tabela 13 – Espaço esperado por entrada da base de dados [Bytes].	30
Tabela 14 – Estimativa do espaço ocupado ao fim de um ano [kBytes].	30

1. Definição do Sistema

No contexto atual de pandemia devido ao COVID-19, muitos são os negócios que viram a sua atividade ser, parcial ou totalmente, transferida para o formato online. Tal prende-se com os longos períodos de confinamento que se têm implementando um pouco por todo o mundo, bem como às elevadas restrições de circulação vigentes.

Portugal não é exceção e são vários os exemplos de negócios em Portugal que se viram forçados a iniciar, melhorar e/ou aumentar a sua presença online. Destaca-se, no entanto, o facto de a sociedade portuguesa não ter um grande historial no que às compras online diz respeito. Dados recentes da comissão europeia demonstram que apenas 44,8% dos portugueses fazem compras online, sendo a média da União Europeia de cerca de 72% (Teixeira, 2020).

Este aspeto, é simultaneamente um obstáculo e uma oportunidade. Obstáculo, pois, muitos negócios não dispõem ainda de qualquer infraestrutura digital, o que obrigará a elevados custos de adaptação. Oportunidade, pois, a margem de mercado é bastante elevada, o que permitirá ganhos de cota àqueles negócios que melhor e mais rápido se adaptarem a operar online.

1.1. Contexto de aplicação do sistema

Em Portugal, no sector do retalho de produtos desportivos, existem três empresas principais, nomeadamente, SportZone, Decathlon e Nike (Ferreira, 2009). No entanto, se se considerar apenas a presença online, verifica-se que apenas a SportZone aparece no pódio. As duas posições restantes são ocupas pelas empresas BoxPT Equipment e Oito.Um. Apesar de estas duas empresas não terem espaços físicos, realizando todo o seu negócio online, a SportZone aparece também como líder no negócio online (Teixeira, 2020).

Neste contexto, verifica-se que existe claramente um líder absoluto no setor do retalho de produtos desportivos em Portugal. No entanto, dado o panorama atual de pandemia, é previsível que muitas das empresas existentes, incluindo a Decathlon e a Nike, tentem aumentar a sua presença online. Poderemos assim, estar a assistir a uma nova era neste setor do retalho e novas configurações no pódio dos maiores retalhistas de produtos desportivos online poderão ocorrer em breve.

1.2. Fundamentação da implementação da base de dados

O presente trabalho de desenvolvimento de uma plataforma de venda online de produtos desportivos foi requisitado por uma empresa, doravante designada cliente, com o perfil PME – pequena e média empresa – que opera em Portugal no sector do retalho.

O cliente pretende fazer face à quebra de vendas que registou durante quase todo o ano de 2020. Para isso, e após um estudo de mercado que permitiu fundamentar essa opção, irá necessitar de criar uma plataforma de e-commerce, que lhe permita transferir parte da sua operação para o canal online.

Uma das componentes essenciais para o sucesso da plataforma a desenvolver consiste numa base de dados onde todos os produtos comercializados pelo cliente se encontrem alojados. No que se segue do presente relatório, serão descritas as principais fases de desenvolvimento da referida base de dados.

Salienta-se desde já o facto de que, pelo seu histórico e perfil, o cliente não pretende fazer uma transição definitiva e em exclusividade para o online, devendo por isso prever-se a operação em simultâneo dos canais tradicional (através da rede de lojas de rua que o cliente tem atualmente) e online.

1.3. Análise da viabilidade do processo

A motivação para o presente trabalho resultou da necessidade de adaptação do cliente ao funcionamento online. Essa adaptação implicará o desenvolvimento de um sistema de gestão de bases de dados, que permita simular no mundo virtual a operação do cliente no mundo real. Para isso, e tendo em conta que o cliente continuará a operar pelo canal tradicional, com o sistema de gestão de base de dados a implementar deverá ser possível:

- manter informação atualizada sobre o stock existente na empresa (inclui o armazém central e as várias lojas físicas). Isto inclui o registo dos produtos existentes, de novos produtos encomendados e dos produtos que vão sendo vendidos;
- saber o estado do relacionamento com os fornecedores da empresa, através do fornecimento de listagens atualizadas com os fornecedores e produtos comprados, bem como variações de preços por estes praticadas;
- controlar em tempo real a faturação da empresa, por loja e produto, de modo a permitir definir as campanhas promocionais de forma mais eficiente que aquela atualmente adotada.

1.4. Estrutura do Relatório

O presente relatório é constituído por um conjunto de seis capítulos. Neste primeiro capítulo apresentam-se o contexto e os objetivos do trabalho desenvolvido. No segundo capítulo, é detalhado o processo de levantamento de requisitos. Segue-se, no capítulo três, a descrição do modelo concetual

que melhor traduz os requisitos previamente estabelecidos. O quarto capítulo aborda os detalhes da conversão do modelo concetual num modelo lógico. Este é por sua vez implementado fisicamente no capítulo cinco. O capítulo seis resume as principais conclusões do trabalho realizado e descrito no presente relatório.

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e de análise de requisitos adotado

O método de levantamento de requisitos foi essencialmente suportado na documentação atualmente existente na empresa. De facto, a maioria dos requisitos anteriormente identificados dizem respeito a entidades que atualmente já fazem parte do ecossistema de operação da empresa.

De forma a complementar a informação existente, foram também consultados alguns documentos existentes na literatura sobre o funcionamento de lojas de e-commerce, com especial enfoque nos produtos desportivos. Pretendeu-se assim perceber algumas das principais particularidades do negócio em formato online, para melhor definir e adaptar os requisitos da base de dados a desenvolver.

Para finalizar, convém referir que a equipa que desenvolveu o presente projeto é utilizadora de plataformas online de compra de produtos desportivos, pelo que o tema não é de toda uma novidade para os autores do trabalho.

2.2. Requisitos levantados

A lista que se segue apresenta, ainda que forma não exaustiva, os requisitos que o novo sistema de gestão de base de dados deverá conseguir dar resposta:

1. O registo de produtos no sistema pressupõe a definição da data de admissão, designação, breve descrição, unidade de medida, quantidade e preço de aquisição. Posteriormente, já na fase de preparação dos produtos para entrada no stock da empresa, deverão ser ainda definidos o valor a que deverão ser vendidos, incluindo o preço unitário e o preço mínimo possível (relevante para definir campanhas promocionais).
2. O sistema deverá permitir definir um número mínimo de unidades de cada produto, após o qual deverá emitir alertas para a encomenda de novas unidades dos produtos com stock inferior ao mínimo definido.
3. Os clientes da loja virtual(armazém) terão de se registar no sistema indicando, pelo menos, nome, morada (rua, localidade e código postal) e email. Opcionalmente poderá complementar

- a sua identificação indicando data de nascimento e telefone(s), em qualquer acesso posterior ao sistema.
4. Os clientes, tanto nas lojas físicas como nas lojas online, poderão associar os seus dados de faturação (NIF) numa compra qualquer (não necessita ser na primeira), de modo a permitir o preenchimento automático de faturas em compras futuras. Não obstante esta situação, a emissão de fatura será uma opção do cliente.
 5. Deverá ser mantido um registo das compras efetuadas por cada cliente de modo a melhor definir campanhas promocionais direcionadas.
 6. As saídas de produtos do sistema deverão ocorrer no momento da sua venda, real ou online, mediante a entrada no sistema do pagamento respetivo, ficando associado o cliente, loja e funcionário.
 7. O sistema deverá ficar com o registo da venda efetuada assim como a sua data processamento, bem como a taxa aplicada aquando da compra.
 8. Os funcionários das lojas reais poderão adicionar observações sobre os clientes de modo a identificar preferências do cliente.
 9. Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por loja e data (intervalo de tempo). Desta forma será possível identificar as lojas mais rentáveis.
 10. Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por cliente, numa determinada data (intervalo de tempo). Desta forma será possível definir campanhas especiais para os melhores clientes.
 11. Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por funcionário e data (intervalo de tempo). Desta forma será possível premiar os funcionários mais produtivos.
 12. Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por produto e data (intervalo de tempo). Desta forma será possível identificar os produtos mais vendidos.
 13. Os clientes poderão classificar a venda atribuindo uma nota e poderão também deixar um comentário.
 14. Para cada loja, deverá ser possível obter o catálogo dos produtos existentes em cada momento.
 15. Para o armazém, deverá ser possível obter o catálogo dos produtos existentes em cada momento.
 16. Deverá ser possível solicitar ao sistema qual o total de vendas de um determinado produto.
 17. Deverá ser possível solicitar ao sistema qual o catálogo de produtos de uma determinada loja.
 18. Deverá ser possível solicitar ao sistema quais os nomes das lojas registadas no sistema.
 19. Deverá ser possível solicitar ao sistema qual o total de faturação numa determinada data.
 20. Deverá ser possível solicitar ao sistema quais as compras realizadas por determinados clientes numa determinada data.
 21. Deverá ser possível solicitar ao sistema quais os valores faturados e produtos que foram vendidos a clientes a partir de uma certa idade.

22. Deverá ser possível solicitar ao sistema quais os tipos de produtos mais vendidos durante uma determinada semana.
23. Deverá ser possível solicitar ao sistema quais os produtos nunca comprados no sistema.
24. Deverá ser possível solicitar ao sistema quais os clientes que nunca compraram nada.
25. Deverá ser possível solicitar ao sistema qual o produto mais comprado, dado pelo nome, quantidade e total faturado.
26. Deverá ser possível solicitar ao sistema quais os melhores Clientes.
27. Deverá ser possível solicitar ao sistema os produtos comprados nos últimos n meses, por ordem de popularidade.
28. As lojas físicas e o armazém central terão horários (dias e horas) de funcionamento variáveis ao longo do tempo, tendo em conta os horários de confinamento definidos nos diferentes municípios onde se localizam. Por sua vez, a loja online funcionará 24/7.
29. As entradas de produtos no sistema deverão ocorrer no momento da sua entrada efetiva no armazém e não no momento da sua encomenda (situação atual). Para isso será necessário registar os produtos que entram no armazém central, no momento da sua receção.
30. Após a venda de um produto, o seu stock deverá ser atualizado descontando as unidades que foram vendidas.
31. Após o fornecimento de um produto, o seu stock deverá ser atualizado incrementando as unidades que foram fornecidas.

2.3. Análise e validação geral dos requisitos

Os requisitos apresentados anteriormente foram validados e aceites pelo cliente, pelo que se passará às fases seguintes. Por forma a facilitar a implementação da base de dados, os requisitos enunciados na secção anterior foram agrupados por tipo de utilização. Assim, nos parágrafos que se seguem apresentam-se os requisitos associados à:

- descrição da base de dados e possíveis vistas de utilização da mesma (Requisitos de Descrição – RD),
- exploração da base de dados e à forma como esta deverá ser povoada e explorada (Requisitos de Exploração – RE);
- controlo da base de dados e aos acessos e privilégios de acesso previstos (Requisitos de controlo – RC).

2.3.1 Requisitos de Descrição

- RD01 - O registo de produtos no sistema pressupõe a definição da data de admissão, designação, breve descrição, unidade de medida, quantidade e preço de aquisição. Posteriormente, já na fase de preparação dos produtos para entrada no stock da empresa,

deverão ser ainda definidos o valor a que deverão ser vendidos, incluindo o preço unitário e o preço mínimo possível (relevante para definir campanhas promocionais).

- RD02 - O sistema deverá permitir definir um número mínimo de unidades de cada produto, após o qual deverá emitir alertas para a encomenda de novas unidades dos produtos com stock inferior ao mínimo definido.
- RD03 - Os clientes da loja virtual(armazém) terão de se registar no sistema indicando, pelo menos, nome, morada (rua, localidade e código postal) e email. Opcionalmente poderá complementar a sua identificação indicando data de nascimento e telefone(s), em qualquer acesso posterior ao sistema.
- RD04 - Os clientes, tanto nas lojas físicas como nas lojas online, poderão associar os seus dados de faturação (NIF) numa compra qualquer (não necessita ser na primeira), de modo a permitir o preenchimento automático de faturas em compras futuras. Não obstante esta situação, a emissão de fatura será uma opção do cliente.
- RD05 - Deverá ser mantido um registo das compras efetuadas por cada cliente de modo a melhor definir campanhas promocionais direcionadas.
- RD06 - As saídas de produtos do sistema deverão ocorrer no momento da sua venda, real ou online, mediante a entrada no sistema do pagamento respetivo, ficando associado o cliente, loja e funcionário.
- RD07 - O sistema deverá ficar com o registo da venda efetuada assim como a sua data processamento, bem como a taxa aplicada aquando da compra.

2.3.2 Requisitos de Exploração

- RE01 - Os funcionários das lojas reais poderão adicionar observações sobre os clientes de modo a identificar preferências do cliente.
- RE02 - Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por loja e data (intervalo de tempo). Desta forma será possível identificar as lojas mais rentáveis.
- RE03 - Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por cliente, numa determinada data (intervalo de tempo). Desta forma será possível definir campanhas especiais para os melhores clientes.
- RE04 - Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por funcionário e data (intervalo de tempo). Desta forma será possível premiar os funcionários mais produtivos.
- RE05 - Deverá ser possível solicitar ao sistema relatórios com os totais de vendas, por produto e data (intervalo de tempo). Desta forma será possível identificar os produtos mais vendidos.
- RE06 - Os clientes poderão classificar a venda atribuindo uma nota e poderão também deixar um comentário.

- RE07 - Para cada loja, deverá ser possível obter o catálogo dos produtos existentes em cada momento.
- RE08 - Para o armazém, deverá ser possível obter o catálogo dos produtos existentes em cada momento.
- RE9 - Deverá ser possível solicitar ao sistema qual o total de vendas de um determinado produto.
- RE10 - Deverá ser possível solicitar ao sistema qual o catálogo de produtos de uma determinada loja.
- RE11 - Deverá ser possível solicitar ao sistema quais os nomes das lojas registadas no sistema.
- RE12 - Deverá ser possível solicitar ao sistema qual o total de faturação numa determinada data.
- RE13 – Deverá ser possível solicitar ao sistema quais as compras realizadas por determinados clientes numa determinada data.
- RE14 - Deverá ser possível solicitar ao sistema quais os valores faturados e produtos que foram vendidos a clientes a partir de uma certa idade.
- RE15 - Deverá ser possível solicitar ao sistema quais os tipos de produtos mais vendidos durante uma determinada semana.
- RE16 - Deverá ser possível solicitar ao sistema quais os produtos nunca comprados no sistema.
- RE17 - Deverá ser possível solicitar ao sistema quais os clientes que nunca compraram nada.
- RE18 - Deverá ser possível solicitar ao sistema qual o produto mais comprado, dado pelo nome, quantidade e total faturado.
- RE19 - Deverá ser possível solicitar ao sistema quais os melhores Clientes.
- RE20 - Deverá ser possível solicitar ao sistema os produtos comprados nos últimos n meses, por ordem de popularidade.

2.3.3 Requisitos de Controlo

- RC01 - As lojas físicas e o armazém central terão horários (dias e horas) de funcionamento variáveis ao longo do tempo, tendo em conta os horários de confinamento definidos nos diferentes municípios onde se localizam. Por sua vez, a loja online funcionará 24/7.
- RC02 - As entradas de produtos no sistema deverão ocorrer no momento da sua entrada efetiva no armazém e não no momento da sua encomenda (situação atual). Para isso será necessário registar os produtos que entram no armazém central, no momento da sua receção.
- RC03 - Após a venda de um produto, o seu stock deverá ser atualizado descontando as unidades que foram vendidas.
- RC04 - Após o fornecimento de um produto, o seu stock deverá ser atualizado incrementando as unidades que foram fornecidas.

3. Modelação Conceptual

Após se terem definido os principais requisitos da base de dados, passou-se para a fase de modelação concetual. Nesta, procurou-se identificar as entidades que constituirão a base de dados, bem como os relacionamentos entre si. Nas secções que se seguem sumariza-se esse processo.

3.1. Identificação e caracterização das entidades

Da análise de requisitos efetuada foi possível identificar as seguintes entidades no sistema:

- **Produto** = {Id + Designacao + Descricao + UnidadeDeMedida + PrecoUnitario + PrecoMinimo}
Contém os dados relativos a cada produto comprado e vendido pela empresa. O dicionário de dados associado a esta entidade engloba os seguintes atributos:
 - Id – chave que identifica inequivocamente cada produto;
 - Designacao – nome do produto;
 - Descricao – pequena descrição relativa ao produto;
 - UnidadeDeMedida – unidade em que a quantidade de produto se encontra definida (ex. un – unidades, kg – quilograma, l – litro);
 - PrecoUnitario – preço base a que o produto deve ser vendido. Este valor tem em conta o custo do produto e a margem de lucro da empresa;
 - PrecoMinimo – valor mínimo a que o produto poderá ser vendido de modo a não trazer prejuízo. Por defeito, este valor deverá corresponder ao maior valor a que cada produto foi comprado. Ou seja, é um valor que deverá ser atualizado a cada nova encomenda de cada produto.
- **Cliente** = {Id + Nome + Email + DataNascimento + NIF + {Telefone} + Endereco{Localidade,CodigoPostal,Rua} + Observacoes}
Contém os dados relativos a cada cliente. O dicionário de dados associado a esta entidade engloba os seguintes atributos:
 - Id – chave que identifica inequivocamente cada cliente;
 - Nome – nome do cliente;
 - Email – email do cliente;
 - DataNascimento – data de nascimento do cliente;
 - NIF – número Identificação Fiscal do cliente;

- Telefone – números de telefone do cliente;
- Endereco – morada do cliente;
- Localidade – localidade do cliente;
- CodigoPostal – código postal do cliente;
- Rua – rua do cliente;
- Observacoes – informações adicionais sobre o cliente.
- Fornecedor = {Id + Nome}

Contém os dados relativos a cada fornecedor de produtos à empresa. O dicionário de dados associado a esta entidade engloba os seguintes atributos:

 - Id – chave que identifica inequivocamente cada fornecedor;
 - Nome – nome do fornecedor.
- Funcionário = {Id + Nome}

Contém os dados relativos a cada funcionário da empresa. O dicionário de dados associado a esta entidade engloba os seguintes atributos:

 - Id – chave que identifica inequivocamente cada funcionário;
 - Nome – nome do funcionário.
- Compra = {Id + Data + CustoTotal + Avaliacao{Nota,Comentario}}
- Contém os dados relativos a cada compra efetuada. O dicionário de dados associado a esta entidade engloba os seguintes atributos:
 - Id Chave que identifica inequivocamente cada compra;
 - Data Data em que foi realizada a compra;
 - CustoTotal Custo total da compra;
 - Tipo Atributo que identifica o tipo loja onde é feita a compra (online ou física);
 - Avaliacao Avaliação da compra feita por um utilizador:
 - Nota Nota que o cliente atribui à compra;
 - Comentario Comentário opcional feito pelo cliente relativo à compra.
- Loja = {Id + Nome + {Telefone} + Tipo}

Contém os dados relativos a cada loja. O dicionário de dados associado a esta entidade engloba os seguintes atributos:

 - Id Chave que identifica inequivocamente cada loja;
 - Nome Nome da loja;
 - TelefoneNúmeros de telefone da loja;
- Armazém = {Id + Nome}

Contém os dados relativos ao armazém. O dicionário de dados associado a esta entidade engloba os seguintes atributos:

 - Id – Chave que identifica inequivocamente o armazém;
 - Nome – Nome do armazém.

3.2. Identificação e caracterização dos relacionamentos

Tendo as entidades definidas na secção anterior, segue-se a definição dos tipos de relacionamentos previstos entre entidades, na fase de exploração da base de dados. De forma a mais facilmente estruturar a definição e caracterização dos relacionamentos, a Tabela 1 resume o processo realizado.

Entidade	Cardinalidade	Relação	Cardinalidade	Entidade
Cliente	1	faz	N	Compra
Compra	N	contém	N	Produto
Funcionário	1	registra	N	Compra
Produto	N	fornecido	N	Fornecedor
Armazém	1	tem	N	Produto
Loja	N	tem	N	Produto
Compra	N	efetuada	1	Loja

Tabela 1 – Relacionamentos existentes na base de dados.

De entre os relacionamentos identificados, verifica-se que alguns apresentam atributos associados. Assim, importa também caracterizar esses atributos.

- Contem (Tabela 'ProdutoCompra' no modelo lógico) = {Taxa, Quantidade, PrecoUnitario}
Contém os dados relativos ao relacionamento contem. O dicionário de dados associado a este relacionamento engloba os seguintes atributos:
 - Taxa – taxa aplicada ao produto no ato da compra;
 - Quantidade – quantidade do produto que foi comprada;
 - PrecoUnitario – preço por unidade do produto.
- Fornecido (Tabela 'ProdutoFornecedor' no modelo lógico) = {Id, Data, Quantidade, PrecoUnitario}
Contém os dados relativos ao relacionamento contem. O dicionário de dados associado a este relacionamento engloba os seguintes atributos:
 - Id – identificador do fornecimento para manter registo dos vários fornecimentos do mesmo produto;
 - Data – Data em que o produto é fornecido;
 - Quantidade – Quantidade do produto que foi fornecida;
 - PrecoUnitario – Preço por unidade do produto.
- Tem = {PontoEncomenda, Stock}
Contém os dados relativos ao relacionamento tem entre as entidades armazém e produto. O dicionário de dados associado a este relacionamento engloba os seguintes atributos:

- Stock – quantidade de cada produto existente em cada momento no sistema (inclui apenas armazém);
 - PontoEncomenda - número mínimo de unidades de cada produto.
 - Tem (Tabela 'ProdutoLoja' no modelo lógico) = {Id, Stock}
- Contém os dados relativos ao relacionamento tem entre as entidades produto e loja. O dicionário de dados associado a este relacionamento engloba os seguintes atributos:
- Id – identificador da relação tem entre produto e loja;
 - Stock – quantidade de cada produto existente em cada momento no sistema (inclui apenas lojas).

3.3. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

Da análise de requisitos efetuada foi possível identificar um conjunto de atributos para as entidades do sistema. Assim, nas Tabelas que se seguem apresenta-se um resumo dos atributos que as várias entidades e relacionamentos do sistema terão de ter.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente cada produto	INTEGER	1..N	Não	Auto incrementado	Simple
Designacao	Nome do produto	VARCHAR(50)	Sequência palavras	Não	""	Simple
Descricao	Pequena descrição relativa ao produto	VARCHAR(100)	Sequência palavras	Não	""	Simple
UnidadeDeMedida	Unidade em que a quantidade de produto se encontra definida (ex. un – unidades, kg – quilograma, l – litro)	VARCHAR(10)	Sequência caracteres	Não	""	Simple
PrecoUnitario	Preço base a que o produto deve ser vendido. Este valor tem em conta o custo do produto e a margem de lucro da empresa.	DOUBLE(8,2)	0.00..N	Não	0.00	Simple
PrecoMinimo	Valor mínimo a que o produto poderá ser vendido de modo a não trazer prejuízo. Por defeito, este valor deverá corresponder ao maior valor a que cada produto foi comprado. Ou seja, é um valor que deverá ser atualizado a cada nova encomenda de cada produto.	DOUBLE(8,2)	0.00..N	Não	0.00	Simple

Tabela 2 – Associação dos atributos com a entidade Produto.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente cada cliente	INTEGER	1..N	Não	Auto incrementado	Simple
Nome	Nome do cliente	CHAR(75)		Não	""	Simple
Email	Email do cliente	CHAR(75)	Sequência de caracteres seguida por "@" seguida pelo domínio do provedor de serviço de email	Não	""	Simple
DataNascimento	Data de nascimento do cliente	DATETIME	Formato ISO 8601 (YYYY-MM-DD)	Sim		Simple
NIF	Número Identificação Fiscal do cliente	INTEGER	Número inteiro com 9 dígitos	Sim		Simple
Telefone	Números de telefone do cliente	CHAR(20)		Sim		Multivalor
Endereco	Morada do cliente			Não		Composto
Localidade	Localidade do cliente	CHAR(75)	Sequência de palavras	Não	""	Simple
CodigoPostal	Código postal do cliente	CHAR(8)	Sequência 4 dígitos numéricos seguidos por "-"	Não	"0000-000"	Simple
Rua	Rua do cliente	CHAR(75)	Sequência de palavras	Não	""	Simple
Observacao	Informações adicionais sobre o cliente	CHAR(500)	Sequência de palavras	Sim	""	Simple

Tabela 3 – Associação dos atributos com a entidade Cliente.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente cada fornecedor	INTEGER	1..N	Não	Auto incrementado	Simple
Nome	Nome do fornecedor	VARCHAR(75)	Sequência palavras	Não	""	Simple

Tabela 4 – Associação dos atributos com a entidade Fornecedor.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente cada funcionário	INTEGER	1..N	Não	Auto incrementado	Simple
Nome	Nome do funcionário	VARCHAR(75)	Sequência palavras	Não	""	Simple

Tabela 5 – Associação dos atributos com a entidade Funcionário.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente cada compra	INTEGER	1..N	Não	Auto incrementado	Simples
Data	Data em que foi realizada a compra	DATETIME	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])	Não	Data corrente	Simples
CustoTotal	Custo total da compra	DOUBLE(8,2)	0.00..N	Não	0.00	Simples
Avaliacao	Avaliação da compra feita por um utilizador			Sim	NULL	Composto
Nota	Nota que o cliente atribui à compra	INTEGER	0..5	Não	0..5	Simples
Comentario	Comentário opcional feito pelo cliente relativo à compra	CHAR(500)	Sequência palavras	Sim	NULL	Simples

Tabela 6 – Associação dos atributos com a entidade Compra.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente cada loja	INTEGER	1..N	Não	Auto incrementado	Simples
Nome	Nome da loja	VARCHAR(75)		Não	""	Simples
Telefone	Números de telefone da loja	CHAR(20)	Sequência de caracteres numéricos começada por "+", seguida de 3 dígitos com indicativo, depois "-", e por último os restantes dígitos.	Sim		Multivalor

Tabela 7 – Associação dos atributos com a entidade Loja.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente o armazém	INTEGER	1..N	Não	Auto incrementado	Simples
Nome	Nome do armazém	VARCHAR(75)	Sequência palavras	Não	""	Simples

Tabela 8 – Associação dos atributos com a entidade Armazém.

De igual modo, também foi possível verificar que alguns dos relacionamentos entre entidades, identificados na secção anterior, só funcionarão como pretendido se tiverem alguns atributos que lhes confirmem funcionalidade adicional. Nas Tabelas que se seguem apresenta-se a caracterização dos atributos que compõem esses relacionamentos.

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Taxa	Taxa aplicada ao produto no ato da compra	DOUBLE(3,2)	0.00..1.00	Não	0.24	Simples
Quantidade	Quantidade do produto que foi comprada	INTEGER	1..N	Não	1	Simples
PrecoUnitario	Preço por unidade do produto	DOUBLE (8,2)	0.00..N	Não	0.00	Simples

Tabela 9 – Associação dos atributos com o relacionamento Contem (Produto-Compra).

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Id	Chave que identifica inequivocamente cada fornecimento	INTEGER	1..N	Não	Auto incrementado	Simples
Data	Data em que o produto é fornecido	DATETIME	Formato ISO 8601 (YYYY-MM-DDThh:mm:ss[.mmm])	Não	Data corrente	Simples
Quantidade	Quantidade do produto que foi fornecida	INTEGER	1..N	Não	1	Simples
PrecoUnitario	Preço por unidade do produto	DOUBLE (8,2)	0.00..N	Não	0.00	Simples

Tabela 10 – Associação dos atributos com o relacionamento Fornecido (Produto-Fornecedor).

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Stock	Quantidade de cada produto existente em cada momento no sistema (inclui armazém e lojas)	INTEGER	0..N	Não	0	Simples
PontoEncomenda	Stock mínimo de cada produto. Se atingido este valor, o sistema deverá alertar para a necessidade de encomendar mais	INTEGER	0..N	Não	0	Simples

Tabela 11 – Associação dos atributos com o relacionamento Tem (Produto-Armazém).

Atributo	Descrição	Domínio	Valores	Nulos	Omissão	Tipo
Stock	Quantidade de cada produto existente em cada momento no sistema (inclui armazém e lojas)	INTEGER	0..N	Não	0	Simples

Tabela 12 – Associação dos atributos com o relacionamento Tem (Produto-Loja).

3.4. Detalhe ou generalização de entidades

O sistema desenvolvido, conforme pensado, não careceu de especializações ou generalizações de entidades. De facto, todas as entidades definidas são “atómicas”, no sentido em que não se prevê a sua subdivisão. Ainda assim, haverá sempre margem no futuro para a distinção de tipos de funcionários em função do tipo de privilégios de uso do sistema, por exemplo.

3.5. Apresentação e explicação do diagrama ER

A informação que foi apresentada ao longo das secções anteriores é bastante completa, em termos da informação que a futura base de dados deverá apresentar. No entanto, aquela apresentação tabular afigura-se bastante desintegrada, não permitindo por isso ter uma perceção global do sistema. Assim, na Figura 1 apresenta-se a informação das tabelas referidas, organizada num diagrama conhecido por diagrama de entidades-relacionamentos, ou simplesmente diagrama ER.

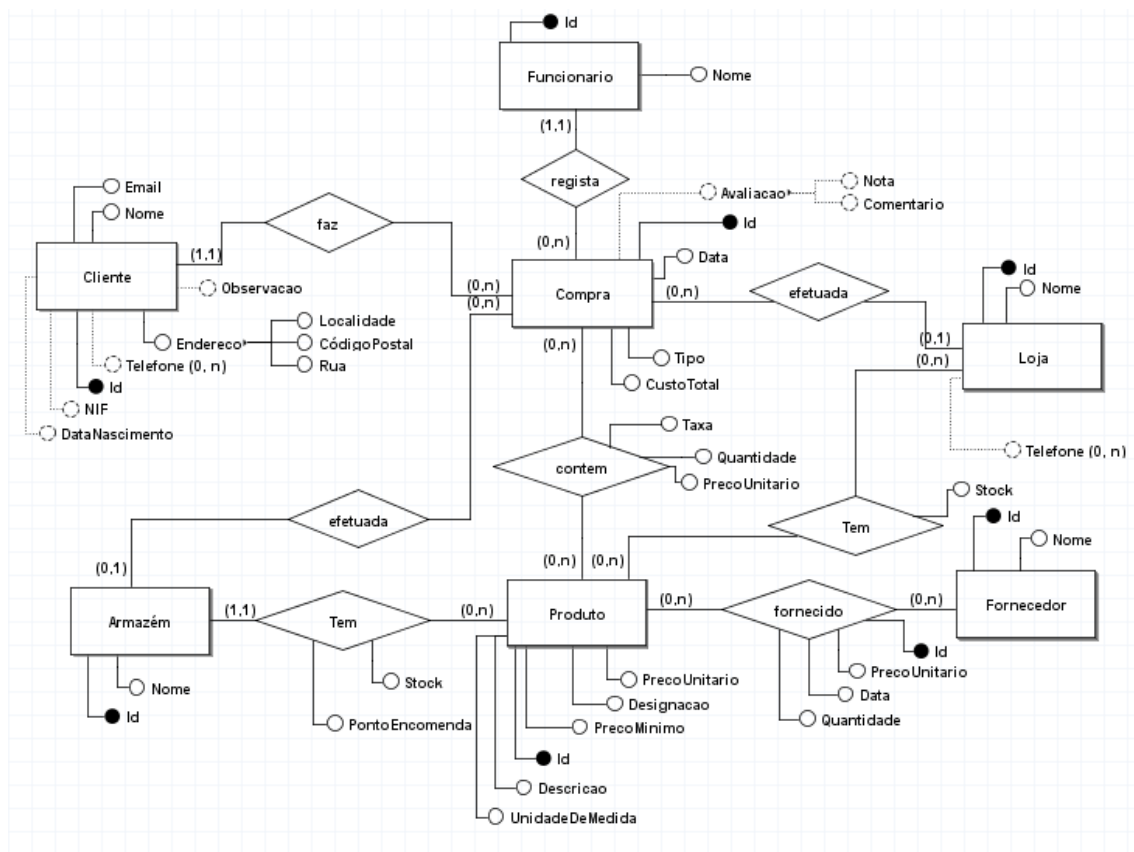


Figura 1 – Diagrama ER do sistema.

3.6. Validação do modelo de dados produzido

A fase de validação do modelo produzido é de suma importância. É nesta fase que se verifica se o modelo produzido é capaz de dar resposta ao conjunto de requisitos definidos pelo cliente. Destaca-se o facto de todo o processo até aqui desenvolvido ser de carácter iterativo. Assim, foi necessário visitar algumas das etapas anteriores algumas vezes antes de se ter chegado ao modelo apresentado na secção anterior.

Tendo em conta todo o processo realizado, o modelo definido no diagrama ER apresentado anteriormente foi considerado validado e adequado para os requisitos definidos. Pode-se assim avançar para as etapas seguintes com confiança.

Nos parágrafos que seguem apresenta-se a demonstração para alguns dos requisitos previamente apresentados na secção 2.2.

1- Os clientes da loja virtual(armazém) terão de se registar no sistema indicando, pelo menos, nome, morada (rua, localidade e código postal) e email. Opcionalmente poderá complementar a sua identificação indicando data de nascimento e telefone(s), em qualquer acesso posterior ao sistema.

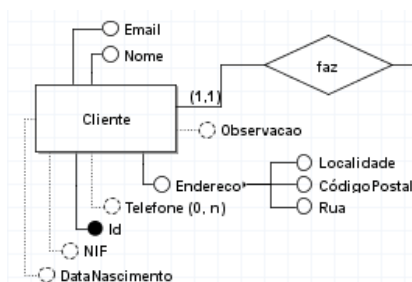


Figura 2 – A entidade Cliente guarda todas as informações referidas.

2- O sistema deverá ficar com o registo da venda efetuada assim como a sua data processamento, bem como a taxa aplicada aquando da compra para cada produto.

A compra tem associada uma data de processamento. A taxa bem como a quantidade e preço unitário de cada produto de uma venda irão ser guardados numa tabela. Esta nova tabela irá ter as chaves estrangeiras das entidades Produto e Compra, e uma compra irá conhecer os produtos e respetivas taxas. O requisito irá ser satisfeito.

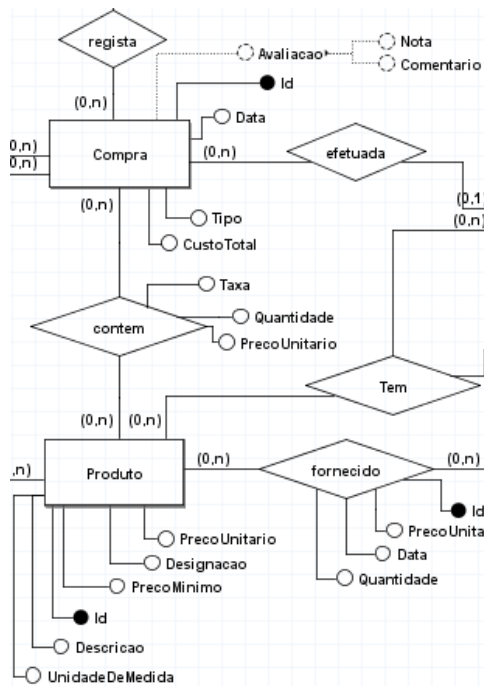


Figura 3 – Diagrama ER do sistema (zoom no relacionamento compra-produto).

3- Para cada loja, deverá ser possível obter o catálogo dos produtos existentes em cada momento.

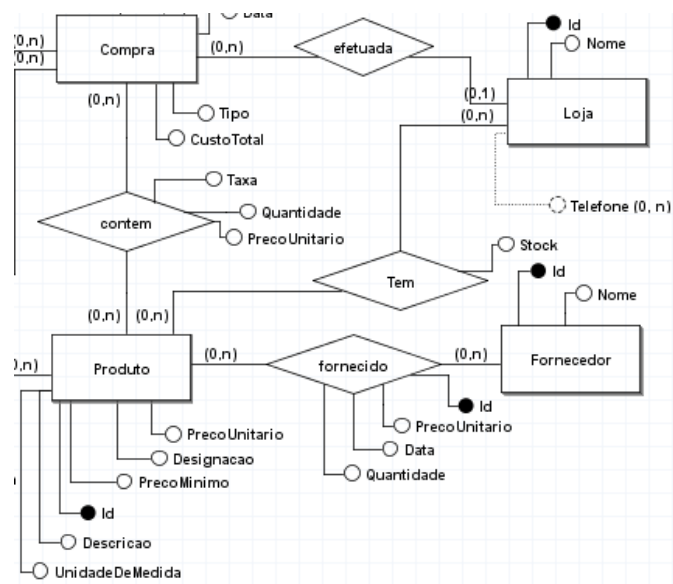


Figura 4 – Diagrama ER do sistema (zoom no relacionamento loja-produto).

Loja e Produto estão relacionados na ordem N:M pelo que cada loja irá ter acesso ao conjunto de produtos em stock. O requisito irá ser satisfeito.

4- Deverá ser possível solicitar ao sistema quais os produtos nunca comprados.

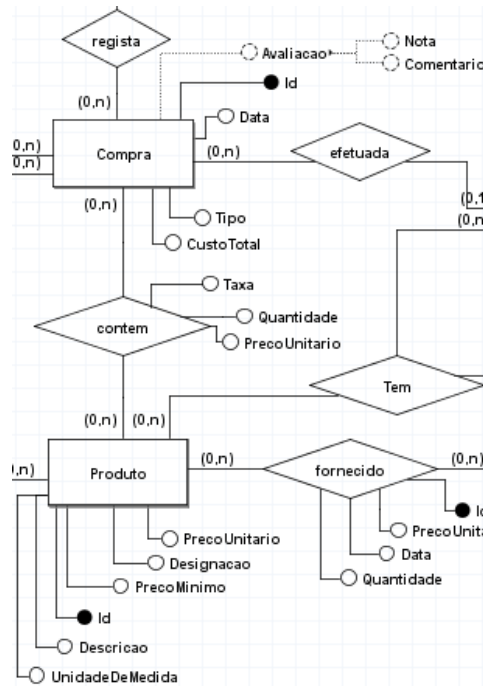


Figura 5 – Diagrama ER do sistema (zoom no relacionamento compra-produto).

Cada compra irá guardar os produtos associados. Desta forma será possível obter os produtos que nunca foram comprados. O requisito é satisfeito.

5- Deverá ser possível solicitar ao sistema quais as compras realizadas por determinados clientes numa determinada data.

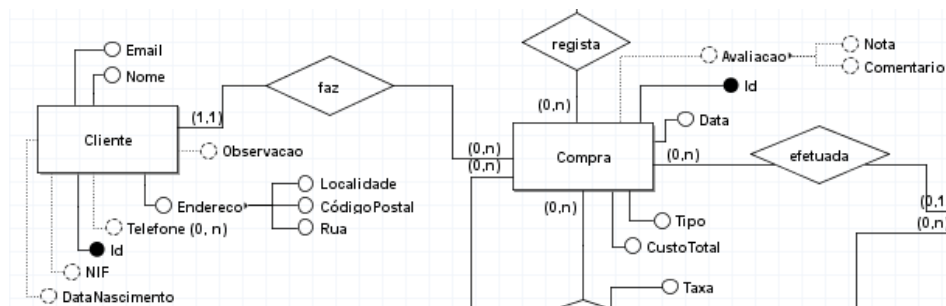


Figura 6 – Diagrama ER do sistema (zoom no relacionamento cliente-compra).

A Compra guarda a sua data de processamento. Como a Compra se relaciona com Cliente na ordem 1:N então irá ser possível conhecer os clientes que fizeram uma determinada compra (cada compra irá ter uma chave estrangeira que identifica quem a fez), numa determinada data. O requisito é satisfeito.

6- Deverá ser possível solicitar ao sistema quais os valores faturados e produtos que foram vendidos a clientes a partir de uma certa idade.

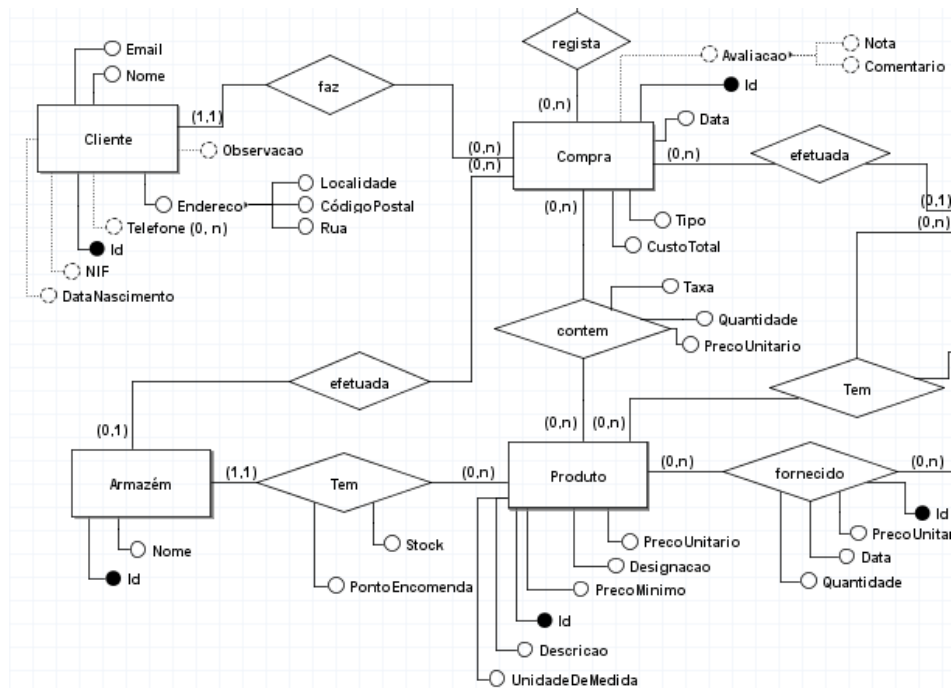


Figura 7 – Diagrama ER do sistema (zoom).

É possível identificar cada cliente a partir de uma compra. Cada cliente pode ter ainda um atributo que guarda a sua data de nascimento. Como Compra e Produto estão relacionados, é possível identificar os diversos produtos que constituem uma compra. A compra possui ainda um custo total. Será assim possível satisfazer o requisito.

4. Modelação Lógica

Concluída e validada a fase de desenvolvimento do modelo concetual, pode afirmar-se que os requisitos do cliente foram atendidos. Assim, prosseguiu-se o trabalho com o desenvolvimento do modelo lógico do sistema de base de dados.

4.1. Construção e validação do modelo de dados lógico

Para a construção do modelo de dados lógicos foram seguidas as regras de conversão do modelo concetual num modelo lógico. Em primeiro lugar, foram identificados as entidades e os seus respetivos atributos. Seguiu-se a derivação dos relacionamentos do modelo de dados relacional.

Segue a explicação do processo de derivação dos relacionamentos, nos pontos 1 e 2. Por fim, no ponto 3, apresentam-se as entidades finais que constituirão o modelo lógico.

1. Relacionamento binário de grau 1:N com participação obrigatória do lado N:

São necessárias duas entidades lógicas, uma para cada entidade, e a chave primária do lado 1 tem de ser usada como atributo na entidade correspondente à entidade do lado N.

Ocorrências:

- 1.1. “faz” entre Cliente(1) e Compra(N)
- 1.2. “regista” entre Funcionario(1) e Compra(N)
- 1.3. “efetuada” entre Loja(1) e Compra(N)
- 1.4. “tem” entre Armazem(1) e Produto(N)
- 1.5. “efetuada” entre Armazem(1) e Compra(N)

2. Relacionamento binário de grau N:M:

São sempre necessárias três entidades lógicas neste tipo de relacionamentos, uma para cada entidade e uma para o relacionamento. As chaves primárias das entidades têm de ser atributos na entidade lógica do relacionamento:

Ocorrências:

- 2.1. “fornecido” entre Produto(N) e Fornecedor(M)
- 2.2. “tem” entre Produto(N) e Loja(M)
- 2.3. “contem” entre Produto(N) e Compra(M)

3. Entidades resultantes:

- Cliente = {Id, Nome, Email, Localidade, CodigoPostal, Rua, NIF, DataNascimento, Observação}
- Produto = {Id, Designacao, Stock, PrecoUnitario, PrecoMinimo, PontoEncomenda, Descricao, UnidadeDeMedida, fk_Armazem_Id}
- Funcionario = {Id, Nome}
- Loja = {Id, Nome}
- Armazem = {Id, Nome}
- Fornecedor = {Id, Nome}
- Compra = {Id, Nota, Comentario, Data, CustoTotal, Tipo, fk_Armazem_Id, fk_Funcionario_Id, fk_Loja_Id, fk_Cliente_Id}
- TelefoneCliente = {Telefone, fk_Cliente_Id}
- TelefoneLoja = {Telefone, fk_Loja_Id }
- ProdutoFornecedor = {Id, fk_Produto_Id, fk_Fornecedor_Id, PrecoUnitario, Data, Quantidade}
- ProdutoLoja = {Stock, fk_Loja_Id, fk_Produto_Id}
- ProdutoCompra = {fk_Produto_Id, fk_Compra_Id, Taxa, Quantidade, PrecoUnitario}

4.2. Desenho do modelo lógico

Tendo em conta todas as considerações tecidas nas secções anteriores acerca da conversão do modelo concetual para lógico, foi desenhado o modelo lógico conforme ilustrado na Figura 8.

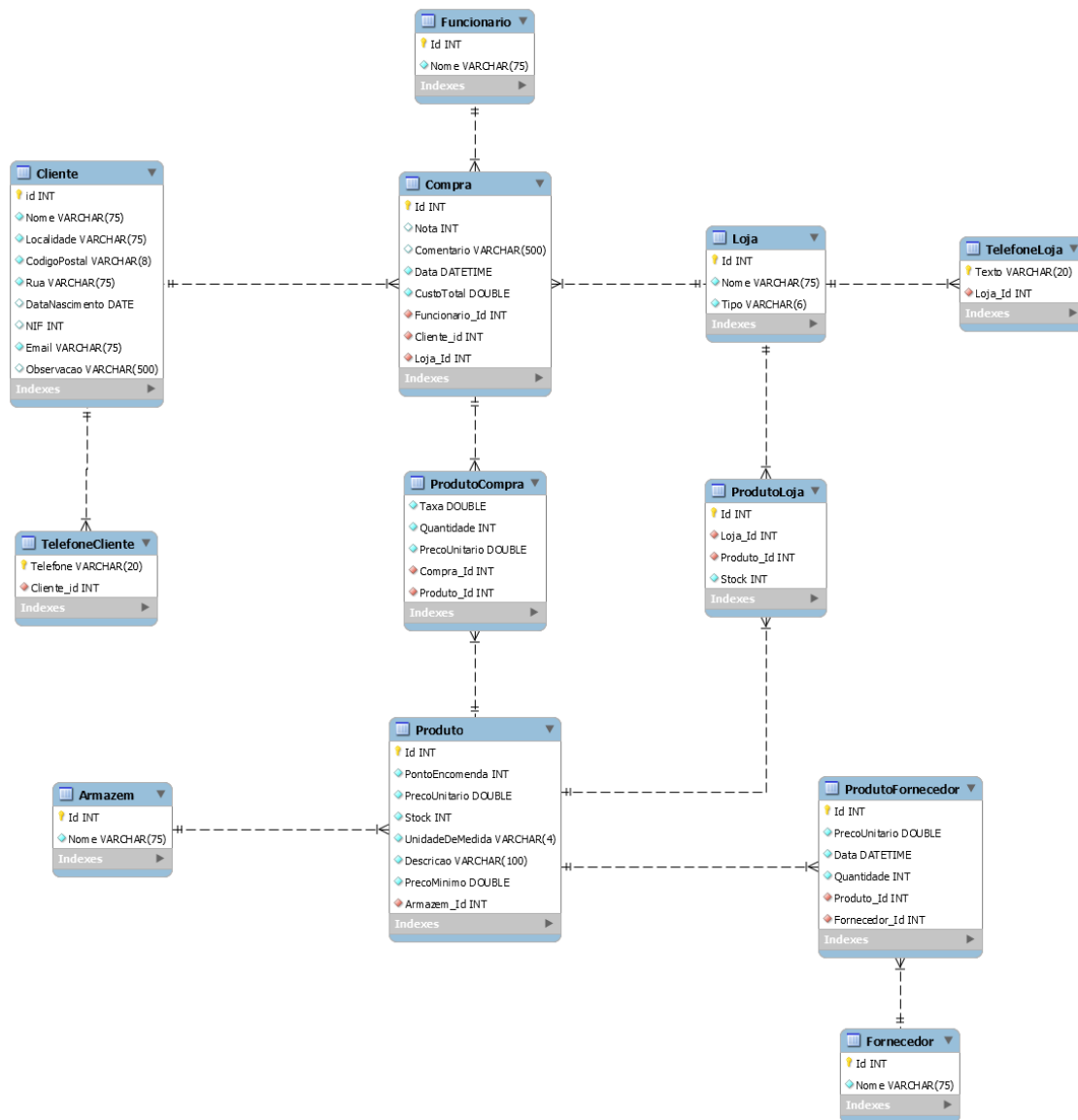


Figura 8 – Modelo lógico do sistema.

4.3. Validação do modelo através da normalização

A normalização consiste num processo que analisa, através de normas, os atributos de uma entidade, de modo a identificar dependências funcionais. O objetivo é minimizar a redundância e inconsistência de dados, facilitando manipulações no banco de dados. Para validar um modelo através da normalização, verificou-se o seguinte conjunto de requisitos:

1. Identificação das dependências funcionais (DF)
2. 1ª forma normal (1FN)
3. 2ª forma normal (2FN)
4. 3ª forma normal (3FN).

- **Dependências funcionais:** no modelo lógico definido nas secções anteriores todas as tabelas apresentam apenas uma chave primária. Assim, não são exetáveis dependências funcionais.
- Uma relação está na **1ª forma normal (1FN)** quando os domínios de todos os atributos consistem apenas em valores atómicos, ou seja, não existem subgrupos de atributos repetidos. O modelo lógico definido anteriormente vai de acordo com esta norma pois nenhum atributo expõe multivalores nem subgrupos de atributos, ou seja, cada linha apresenta apenas um valor por coluna. Por exemplo, o atributo multivalorado TelefoneCliente na entidade Cliente deu origem a uma nova tabela no modelo lógico desenvolvido.
- Uma tabela está na **2ª Forma Normal (2FN)** se respeitar a 1FN e se todos os atributos não chave forem totalmente dependentes da chave primária. Mais uma vez, o modelo desenvolvido vai de acordo com a 2FN pois respeita a 1FN e todos os atributos não chave das tabelas que o constituem, são totalmente dependentes da chave primária, não existindo dependências parciais que possam causar redundância de informação.
- Uma tabela está na **3ª Forma Normal (3FN)** se respeitar a 2FN e se nenhum atributo não chave é transitivamente depende da chave primária. Também este critério se verifica no modelo desenvolvido. Este vai de encontro com a 2FN e nenhum dos seus atributos admite dependências transitivas pois em todos os relacionamentos que existem só há dependência da chave primária.

4.4. Validação do modelo com interrogações do utilizador

A validação do modelo foi realizada com base em algumas *queries* definidas em conjunto com o cliente. Durante a discussão da forma como essas *queries* funcionariam foram levantadas mais algumas dúvidas por parte do cliente. Tendo essas dúvidas sido esclarecidas, considerou-se o modelo validado pelo cliente.

4.5. Reavaliação do modelo lógico (se necessário)

Embora nas secções anteriores se apresente a versão final do modelo lógico, a verdade é que o processo até se obter aquela versão foi iterativo. Assim, foi necessário por vezes rever algumas das opções tomadas de modo a obter a solução final encontrada. Nesta fase, admitiu-se que o modelo se encontra estabilizado pelo que não foi necessário reavaliar o modelo lógico.

4.6. Revisão do modelo lógico produzido

Após o desenvolvimento do modelo lógico deverá ser importante reunir novamente com o cliente para aprovação e validação do modelo e para que se possa avançar para a próxima fase do projeto.

Dada a complexidade do modelo lógico, será necessário clarificar e analisar alguns traços gerais deste tipo de modelos para que o cliente perceba e confirme que o modelo proposto atende às necessidades do mesmo.

Depois de validar as entidades presentes no modelo conceptual, o cliente verificou que os atributos de todas as entidades coincidiam com o pretendido de forma a guardar e armazenar toda a informação requerida. Posteriormente, em conjunto com o cliente, foram formuladas algumas questões para verificar que a estrutura do modelo era capaz de todas as solicitações de forma clara e simples.

Cumpridas todas as satisfações do cliente, este deu autorização para avançar para a próxima fase do projeto.

5. Implementação Física

O conjunto de etapas que compõe a implementação física do modelo lógico apresentado no capítulo anterior são descritas nas secções que se seguem.

5.1. Seleção do sistema de gestão de bases de dados

Nesta fase do projeto, foi preciso escolher um Sistema de Gestão de Base de Dados (SGBD) que se adaptasse às necessidades da base de dados desenvolvida e que fosse uma boa opção tendo em conta a manutenção futura. O SGBD Relacional selecionado para implementar a BD descrita nos capítulos anteriores foi o MySQL da Oracle (escolha influenciada pela proximidade à UC de Base de Dados, facilitando a adaptação e aprendizagem do Sistema).

Para além desta razão, o InnoDB (sistema de armazenamento e gestão de transações do MySQL), segue os princípios ACID (Atomicidade, Consistência, Isolamento e Durabilidade), o que também pesou na escolha. Por outro lado, a utilização do MySQL Workbench, uma ferramenta visual que permite alterar e desenvolver uma base de dados, é facilitada através do comando Forward Engineering capaz de desenhar um diagrama ER e rapidamente o projetar numa base de dados física capaz de ser manipulada.

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

O SGBD escolhido (MySQL) tem uma aplicação (MySQL Workbench) que permite a passagem de um diagrama/modelo lógico para uma base de dados relacional e efetuar operações que em outros sistemas de gestão de base de dados, levariam algum tempo e esforço. Assim o comando Forward Engineering permite automatizar o processo top-down de construção de componentes de baixo nível, neste caso a implementação no SGBD, através de uma abstração de alto nível (o modelo lógico). Esta opção cria automaticamente as tabelas com os seus atributos e chaves e executa-a, criando o modelo físico da base de dados.

5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)

Para esta etapa foram usadas algumas ações (designadas queries no contexto das bases de dados) que pretendem dar resposta a alguns requisitos definidos no capítulo 2.

- Query 1: Qual o total de vendas do Produto com Id = 1?

```
SELECT SUM(Quantidade) AS Total FROM ProdutoCompra WHERE Produto_Id = 1;
```

- Query 2: Qual o catalogo de produtos da Loja com Id = 3?

```
SELECT DISTINCT(Descricao) AS Catalogo FROM ProdutoLoja  
JOIN Produto ON Loja_Id = 3;
```

- Query 3: Quais os nomes das lojas registadas no sistema?

```
SELECT Nome AS Nomes FROM Loja;
```

- Query 4: Qual o total de faturação durante o mês de março de 2020?

```
SELECT SUM(CustoTotal) AS Faturacao FROM Compra  
WHERE `Data` BETWEEN '2020-03-01 00:00:00' AND '2020-03-31 23:59:59';
```

- Query 5: Qual é o top 10 clientes do Sistema, dado pelo nome e total de compras associado?

```
SELECT Nome ,COUNT(Cliente_Id) AS TotalCompras FROM Cliente  
JOIN Compra ON Compra.Cliente_Id = Cliente.Id  
GROUP BY Cliente.Id  
ORDER BY TotalCompras DESC  
LIMIT 10;
```

- Query 6: Quais as compras realizadas pelos clientes com id = 1, 2 e 3 durante o mes de Março de 2020?

```
SELECT Cliente.Nome FROM Cliente  
JOIN Compra ON Cliente.id = Compra.Cliente_Id  
WHERE Cliente.id >=1 AND Cliente.id<=3 AND year(Compra.`Data`)=2020 AND month(Compra.`Data`)=03;
```

- Query 7: Quais os valores faturados e produtos que foram vendidos aos clientes com mais de 40 anos?

```
SELECT DISTINCT P.Descricao, PC.Quantidade * PC.PrecoUnitario AS TOTAL FROM Produto AS P
JOIN ProdutoCompra AS PC ON P.id=PC.Produto_Id
JOIN Compra AS C ON PC.Compra_Id = C.Id
JOIN Cliente AS Cl ON C.Cliente_id = Cl.Id
WHERE year(NOW())-year(Cl.DataNascimento)>=40;
```

- Query 8: Quais os tipos dos 5 produtos mais vendidos durante a semana 15 de 2020?

```
SELECT DISTINCT P.Descricao, SUM(PC.Quantidade) AS Soma FROM Produto AS P
JOIN ProdutoCompra AS PC on P.Id = PC.Produto_Id
JOIN Compra AS C on PC.Compra_Id = C.Id
WHERE week(C.`Data`) =15 AND year(C.`Data`)=2020
group by P.Id
order by Soma desc
LIMIT 5;
```

- Query 9: Quais os produtos nunca comprados?

```
SELECT P.Id, P.Descricao FROM Produto AS P
LEFT JOIN ProdutoCompra as PC ON P.Id = PC.Produto_Id
WHERE PC.Compra_Id is null;
```

- Query 10: Quais os clientes que nunca compraram nada?

```
SELECT Cl.id, Cl.Nome FROM Cliente AS Cl
WHERE Cl.id NOT IN (SELECT Cl.id FROM Cliente AS Cl
JOIN Compra AS C ON Cl.id=C.Cliente_id
JOIN ProdutoCompra AS PC ON C.Id = PC.Produto_Id);
```

- Query 11: Qual o produto mais comprado no sistema, dado pela quantidade comprada e o total faturado?

```
SELECT P.Descricao, PC.Quantidade, SUM(PC.Quantidade*PC.PrecoUnitario) AS Valor FROM Produto AS P
JOIN ProdutoCompra AS PC ON P.Id = PC.Produto_Id
JOIN Compra as C ON PC.Compra_id = C.Id
GROUP BY P.Descricao;
```

- Procedure 1: Qual o produto mais comprado no sistema, dado pela quantidade comprada e o total faturado, no mês n?

```

delimiter $$
CREATE PROCEDURE produtosPopularesUltimosMeses(n Int)
begin
    SELECT P.Descricao, PC.Quantidade, SUM(PC.Quantidade*PC.PrecoUnitario) AS Valor FROM Produto AS P
    JOIN ProdutoCompra AS PC ON P.Id = PC.Produto_Id
    JOIN Compra as C ON PC.Compra_id = C.Id
    WHERE month(NOW())- month(C.`Data`) <=n
    GROUP BY P.Descricao
    order by PC.Quantidade desc;
end$$
call produtosPopularesUltimosMeses(2);

```

5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)

Apesar da ferramenta de base de dados mySql originar índices para as chaves primárias de cada tabela, por defeito, achamos por bem introduzir mais um índice com o objetivo de melhorar a performance das nossas *queries* e as tornar mais eficientes.

Como muitas das nossas *queries* fazem a consulta através do atributo 'Data' definida em Compra, achamos por bem criar um índice para o efeito.

```
CREATE INDEX idx_Data ON Compra (`Data`);
```

Caso o volume de dados venha a aumentar futuramente, este índice irá melhorar a procura de compras numa determinada data, operação que esperamos que aconteça com bastante frequência.

Apesar de considerarmos bastante útil a criação de índices, temos também consciência de que a nossa base de dados não apresenta registos suficientes para se notar impacto na eficiência. A criação de mais índices traria um custo adicional não justificável para o volume de dados tratados na nossa base de dados.

5.5. Estimativa do espaço em disco da base de dados e taxa de crescimento anual

Os valores da Tabela 13 foram calculados com base na soma do espaço ocupado por cada atributo que constitui uma tabela.

Atributo	Espaço esperado por entrada {Bytes}
Funcionario	80
Compra	548
Cliente	825

TelefoneCliente	25
ProdutoCompra	28
Produto	138
Armazem	80
Loja	80
TelefoneLoja	80
ProdutoLoja	16
ProdutoFornecedor	32
Fornecedor	80

Tabela 13 – Espaço esperado por entrada da base de dados {Bytes}.

A estrutura da base de dados vai ter um custo de memória inicial que corresponde às tabelas criadas para as entidades loja, armazém, funcionário e fornecedor. Estas tabelas são essencialmente fixas e no contexto geral da base de dados contribuem uma quantidade negligenciável para o espaço total que a mesma ocupa.

Para fazer previsões da quantidade de espaço que tudo o resto vai ocupar vamos precisar de assumir algumas coisas, assim vamos trabalhar com os seguintes parâmetros:

Entidade	Valor Inicial (em unidades)	Espaço Ocupado Inicial (em Kbytes)	Aumento /mes	Valor ao fim do ano	Espaço Ocupado ao fim do ano (em Kbytes)
Cliente	100	80.57	20%	900	725.1
Compra	1/mês cliente	0	1 / cliente	4750	2542
Produto	100	13.48	5%	180	24.26
ProdutoFornecedor	100	3.125	Metade do produtos	935	29.22

Tabela 14 – Estimativa do espaço ocupado ao fim de um ano [kBytes].

As colunas **Valor Inicial** e **Aumento /mês** foram preenchidas com valores fornecidos e estimados pelo cliente.

Os valores da coluna **Espaço Ocupado Inicial** e **Espaço Ocupado ao fim do ano** foram calculados com base na multiplicação do número de entidades e o espaço esperado por entrada.

Se as estimativas dadas forem verdadeiras, a base de dados irá crescer, no máximo, 3.24 MB.

Podemos ver que mesmo entre as tabelas que vão ocupar mais espaço existe uma diferença grande da quantidade de espaço necessária para arquivar as compras e os clientes e tudo o resto. Uma parte grande deste espaço provém dos campos de Observações por isso uma potencial estratégia para o reduzir o peso da base de dados seria reduzir o tamanho desse campo ou potencialmente criar uma tabela nova para guardar Observações apenas nos casos em que estas existem.

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

As vistas são tabelas lógicas criadas em *runtime* quando executadas. Têm especial importância pois ajudam a simplificar *queries* complexas e porque possibilita restringir e delimitar o acesso a dados proporcionando uma camada de segurança extra.

As vistas utilizadas na nossa base de dados foram as seguintes:

- Vista 1: Disponibilizar uma lista com as datas de aniversário e idades dos clientes atuais do sistema. Um exemplo de funcionalidade desta vista será poder premiar clientes nos seus aniversários com descontos.

```
CREATE VIEW vwListaAniversariosIdades AS
SELECT Cliente.DataNascimento, datediff(NOW(),(Cliente.DataNascimento)) div 365 AS Idades FROM Cliente;

SELECT * FROM vwListaAniversariosIdades;
```

- Vista 2: Disponibilizar uma lista com o número total das compras por localidade dos clientes, apresentada por ordem decrescente (do número total de compras). Um exemplo de funcionalidade desta vista será permitir estudar as lojas e zonas mais rentáveis por forma a reforçar o stock ou as menos rentáveis por forma a proporcionarem descontos e campanhas.

```
CREATE VIEW vwListaComprasPorLocalidade AS
SELECT Cl.localidade, COUNT(C.Cliente_Id) AS `Total de Compras` FROM Cliente AS Cl
JOIN Compra AS C ON C.Cliente_Id = Cl.Id
GROUP BY Cl.Id
ORDER BY `Total de Compras` desc;

SELECT * FROM vwListaComprasPorLocalidade;
```

5.7. Revisão do sistema implementado

Esta ultima secção do trabalho é talvez a primeira em grau de importância para o cliente. De facto, é nesta etapa do processo que o cliente tem pela primeira vez contacto com aquela que virá a ser a sua futura ferramenta de trabalho. Assim, o processo de revisão do sistema consistiu numa pequena demonstração das funcionalidades implementadas. Na prática, tratou-se de uma espécie de visita guiada na qual o cliente, guiado pela equipa de desenvolvimento, explorou as potencialidades do sistema.

Foi demonstrado que o sistema conseguia atender a todos os requisitos pré-definidos em conjunto com o cliente, nomeadamente através da execução de algumas *queries* específicas. Também foram testadas outras *queries* entretanto sugeridas pelo cliente.

6. Conclusões e Trabalho Futuro

O presente relatório resumiu as principais etapas do trabalho realizado por um grupo de alunos do Mestrado Integrado em Engenharia Informática. Não obstante o trabalho ser eminentemente académico, foi encarado pelo grupo como uma verdadeira simulação de um cenário real.

Assim, simulou-se a existência de um potencial cliente que estaria interessado no desenvolvimento de um sistema de gestão de bases de dados. Foi pesquisado um contexto atual pertinente e foi criado um cenário bastante realista. Para tal, foi importante a experiência de vida dos membros do grupo no contexto particular em que o presente trabalho se insere.

O problema abordado prendia-se com a necessidade que uma empresa retalhista de produtos desportivos tinha de fazer face aos constrangimentos impostos pelo cenário de pandemia que vivemos hoje em dia. Para isso, teria de passar a operar no mercado online. Tal só seria possível se dispusesse de um sistema online para realizar as suas vendas – vulgo plataforma de e-commerce. Por sua vez, esse sistema teria de ser suportado por uma base de dados, tópico principal do presente relatório.

O trabalho foi dividido em quatro principais tarefas. Primeiro foram identificados um conjunto de requisitos e necessidades para o sistema que a empresa cliente pretendia implementar. Este levantamento foi realizado tendo em conta os limites de tempo/espço do presente trabalho. Assim, a lista de requisitos apresentada não foi exaustiva, apresentando, no entanto, uma dimensão considerada razoável para o contexto. Como melhorias futuras, pode desde já referir-se o aprofundamento do levantamento de requisitos. O restante do trabalho teria naturalmente de ser atualizado em função dos novos requisitos definidos.

Em segundo lugar, foi desenvolvido um primeiro modelo concetual do sistema. Este modelo seria a principal forma de selar o contrato entre o cliente e a equipa de desenvolvimento. É através deste que a equipa de desenvolvimento mostraria que apreendera as necessidades do cliente. Por sua vez, era sobre este que as validações/alterações do cliente seriam mais facilmente definidas. Nesta tarefa em concreto, o grupo está bastante satisfeito com o trabalho realizado. Ficaram identificadas as principais etapas do desenvolvimento de um modelo concetual. Assim, será fácil, mediante à adição de novos requisitos, atualizar o modelo concetual, caso se venha a revelar pertinente.

A terceira grande tarefa consistiu na tradução do modelo concetual desenvolvido anteriormente, num modelo lógico. Esta tarefa incluiu a conversão do modelo concetual, tendo em conta as entidades e os relacionamentos existentes, bem como a natureza de ambos. A normalização e a validação com cenários de utilização reais foram bastante relevantes nesta tarefa. De facto, durante essas etapas verificou-se

por vezes a necessidade de rever alguns detalhes no modelo definido. Isso acabou por expor o grupo ao processo inerentemente iterativo do desenvolvimento de bases de dados.

A quarta e última grande tarefa traduziu-se na implementação do modelo lógico num sistema físico. Ou seja, todo o trabalho realizado até aqui havia sido realizado num ambiente “externo ao computador” (embora se tenham usado várias ferramentas informáticas), sendo apenas nesta quarta tarefa que a base de dados ganha vida informática e passa a ser traduzida por bits. Esta tarefa foi sem dúvida a mais apreciada pelo grupo, pelo que pouco mais haverá a acrescentar. Em todo o caso, o facto de surgir apenas no final, reforça a ideia da complementaridade de capacidades que um engenheiro informático deve possuir, não se limitando apenas ao que a formação de base traz per si.

Tendo em consideração o trabalho desenvolvido, acredita o grupo que seria relativamente pacífica a conversão deste exercício académico para o mundo real. Foi com esse espírito que o trabalho foi encarado desde o início.

Referências Bibliográficas

Teixeira, A., 2020. Como Está A Evoluir O E-Commerce Em Portugal Em 2020 - Digitalks. [online] Digitalks Portugal. Disponível em: <https://digitalks.pt/artigos/a-evolucao-do-e-commerce-em-portugal/> [Acedido 7 Novembro 2020].

Ferreira, P. - A evolução do retalho de artigos desportivos on-line no século XXI: Projecto Sport Zone on-line. Lisboa: ISCTE, 2009. Dissertação de mestrado. Disponível em <http://hdl.handle.net/10071/1947> [Acedido 7 Novembro 2020].

Connolly, T., Begg, C., Database Systems: A Practical Approach to Design, Implementation, and Management, AddisonWesley, Global Edition, 26 Sep 2014. ISBN-10: 1292061189, ISBN-13: 978-1292061184.

Lista de Siglas e Acrónimos

BD	<i>Base de Dados</i>
SGBD	<i>Sistema de Gestão de Base de Dados</i>
ACID	<i>Atomicidade, Consistência, Isolamento e Durabilidade</i>
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada em português)
PK	<i>Primary Key</i> (Chave primária em português)
E-R	<i>Entidade – Relacionamento</i>

Anexos

I. Anexo 1 – Script para geração do modelo físico

```
-- 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='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----
-- Schema mydb
-- -----

drop schema if exists SGBD;
create schema SGBD;
use SGBD;

-- -----
-- Table `Cliente`
-- -----

DROP TABLE IF EXISTS `Cliente` ;

CREATE TABLE IF NOT EXISTS `Cliente` (
  `id` INT NOT NULL,
  `Nome` VARCHAR(75) NOT NULL,
  `Localidade` VARCHAR(75) NOT NULL,
  `CodigoPostal` VARCHAR(8) NOT NULL,
  `Rua` VARCHAR(75) NOT NULL,
  `DataNascimento` DATE NULL,
  `NIF` INT NULL,
  `Email` VARCHAR(75) NOT NULL,
  `Observacao` VARCHAR(500) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- -----
```

```

-- Table `TelefoneCliente`
-- -----
DROP TABLE IF EXISTS `TelefoneCliente` ;

CREATE TABLE IF NOT EXISTS `TelefoneCliente` (
  `Telefone` VARCHAR(20) NOT NULL,
  `Cliente_id` INT NOT NULL,
  PRIMARY KEY (`Telefone`),
  CONSTRAINT `fk_TelefoneCliente_Cliente1`
    FOREIGN KEY (`Cliente_id`)
    REFERENCES `Cliente` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `fk_TelefoneCliente_Cliente1_idx` ON `TelefoneCliente`
(`Cliente_id` ASC) VISIBLE;

-- -----
-- Table `Funcionario`
-- -----
DROP TABLE IF EXISTS `Funcionario` ;

CREATE TABLE IF NOT EXISTS `Funcionario` (
  `Id` INT NOT NULL,
  `Nome` VARCHAR(75) NOT NULL,
  PRIMARY KEY (`Id`))
ENGINE = InnoDB;

-- -----
-- Table `Loja`
-- -----
DROP TABLE IF EXISTS `Loja` ;

CREATE TABLE IF NOT EXISTS `Loja` (
  `Id` INT NOT NULL,
  `Nome` VARCHAR(75) NOT NULL,
  PRIMARY KEY (`Id`))
ENGINE = InnoDB;

-- -----
-- Table `Armazem`
-- -----

```

```
DROP TABLE IF EXISTS `Armazem` ;
```

```
CREATE TABLE IF NOT EXISTS `Armazem` (  
  `Id` INT NOT NULL,  
  `Nome` VARCHAR(75) NOT NULL,  
  PRIMARY KEY (`Id`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `Compra`  
-- -----
```

```
DROP TABLE IF EXISTS `Compra` ;
```

```
CREATE TABLE IF NOT EXISTS `Compra` (  
  `Id` INT NOT NULL,  
  `Nota` INT NULL,  
  `Comentario` VARCHAR(500) NULL,  
  `Data` DATETIME NOT NULL,  
  `CustoTotal` DOUBLE NOT NULL,  
  `Funcionario_Id` INT NOT NULL,  
  `Cliente_id` INT NOT NULL,  
  `Loja_Id` INT NULL,  
  `Tipo` VARCHAR(6) NOT NULL,  
  `Armazem_Id` INT NULL,  
  PRIMARY KEY (`Id`),  
  CONSTRAINT `fk_Compra_Funcionario1`  
    FOREIGN KEY (`Funcionario_Id`)  
    REFERENCES `Funcionario` (`Id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Compra_Cliente1`  
    FOREIGN KEY (`Cliente_id`)  
    REFERENCES `Cliente` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Compra_Loja1`  
    FOREIGN KEY (`Loja_Id`)  
    REFERENCES `Loja` (`Id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Compra_Armazem1`  
    FOREIGN KEY (`Armazem_Id`)  
    REFERENCES `Armazem` (`Id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```

ENGINE = InnoDB;

CREATE INDEX `fk_Compra_Funcionario1_idx` ON `Compra` (`Funcionario_Id` ASC)
VISIBLE;

CREATE INDEX `fk_Compra_Cliente1_idx` ON `Compra` (`Cliente_id` ASC) VISIBLE;

CREATE INDEX `fk_Compra_Loja1_idx` ON `Compra` (`Loja_Id` ASC) VISIBLE;

CREATE INDEX `fk_Compra_Armazem1_idx` ON `Compra` (`Armazem_Id` ASC) VISIBLE;

-- -----
-- Table `TelefoneLoja`
-- -----
DROP TABLE IF EXISTS `TelefoneLoja` ;

CREATE TABLE IF NOT EXISTS `TelefoneLoja` (
  `Texto` VARCHAR(20) NOT NULL,
  `Loja_Id` INT NOT NULL,
  PRIMARY KEY (`Texto`),
  CONSTRAINT `fk_TelefoneLoja_Loja1`
    FOREIGN KEY (`Loja_Id`)
    REFERENCES `Loja` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `fk_TelefoneLoja_Loja1_idx` ON `TelefoneLoja` (`Loja_Id` ASC)
VISIBLE;

-- -----
-- Table `Produto`
-- -----
DROP TABLE IF EXISTS `Produto` ;

CREATE TABLE IF NOT EXISTS `Produto` (
  `Id` INT NOT NULL,
  `PontoEncomenda` INT NOT NULL,
  `PrecoUnitario` DOUBLE NOT NULL,
  `Stock` INT NOT NULL,
  `UnidadeDeMedida` VARCHAR(4) NOT NULL,
  `Descricao` VARCHAR(100) NOT NULL,
  `PrecoMinimo` DOUBLE NOT NULL,
  `Armazem_Id` INT NOT NULL,

```

```

        `Designacao` VARCHAR(45) NOT NULL,
PRIMARY KEY (`Id`),
CONSTRAINT `fk_Produto_Armazem1`
    FOREIGN KEY (`Armazem_Id`)
    REFERENCES `Armazem` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `fk_Produto_Armazem1_idx` ON `Produto` (`Armazem_Id` ASC)
VISIBLE;

-- -----
-- Table `ProdutoLoja`
-- -----

DROP TABLE IF EXISTS `ProdutoLoja` ;

CREATE TABLE IF NOT EXISTS `ProdutoLoja` (
    `Loja_Id` INT NOT NULL,
    `Produto_Id` INT NOT NULL,
    `Stock` INT NOT NULL,
CONSTRAINT `fk_Tem_Loja1`
    FOREIGN KEY (`Loja_Id`)
    REFERENCES `Loja` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Tem_Produto1`
    FOREIGN KEY (`Produto_Id`)
    REFERENCES `Produto` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `fk_Tem_Loja1_idx` ON `ProdutoLoja` (`Loja_Id` ASC) VISIBLE;

CREATE INDEX `fk_Tem_Produto1_idx` ON `ProdutoLoja` (`Produto_Id` ASC)
VISIBLE;

-- -----
-- Table `ProdutoCompra`
-- -----

DROP TABLE IF EXISTS `ProdutoCompra` ;

CREATE TABLE IF NOT EXISTS `ProdutoCompra` (

```



```

`Taxa` DOUBLE NOT NULL,
`Quantidade` INT NOT NULL,
`PrecoUnitario` DOUBLE NOT NULL,
`Compra_Id` INT NOT NULL,
`Produto_Id` INT NOT NULL,
CONSTRAINT `fk_Contem_Compra1`
    FOREIGN KEY (`Compra_Id`)
    REFERENCES `Compra` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Contem_Produto1`
    FOREIGN KEY (`Produto_Id`)
    REFERENCES `Produto` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `fk_Contem_Compra1_idx` ON `ProdutoCompra` (`Compra_Id` ASC)
VISIBLE;

CREATE INDEX `fk_Contem_Produto1_idx` ON `ProdutoCompra` (`Produto_Id` ASC)
VISIBLE;

-- -----
-- Table `Fornecedor`
-- -----
DROP TABLE IF EXISTS `Fornecedor` ;

CREATE TABLE IF NOT EXISTS `Fornecedor` (
    `Id` INT NOT NULL,
    `Nome` VARCHAR(75) NOT NULL,
    PRIMARY KEY (`Id`))
ENGINE = InnoDB;

-- -----
-- Table `ProdutoFornecedor`
-- -----
DROP TABLE IF EXISTS `ProdutoFornecedor` ;

CREATE TABLE IF NOT EXISTS `ProdutoFornecedor` (
    `Id` INT NOT NULL,
    `PrecoUnitario` DOUBLE NOT NULL,
    `Data` DATETIME NOT NULL,
    `Quantidade` INT NOT NULL,

```

```

`Produto_Id` INT NOT NULL,
`Fornecedor_Id` INT NOT NULL,
PRIMARY KEY (`Id`),
CONSTRAINT `fk_Fornecido_Produto1`
    FOREIGN KEY (`Produto_Id`)
    REFERENCES `Produto` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Fornecido_Fornecedor1`
    FOREIGN KEY (`Fornecedor_Id`)
    REFERENCES `Fornecedor` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `fk_Fornecido_Produto1_idx` ON `ProdutoFornecedor` (`Produto_Id`
ASC) VISIBLE;

CREATE INDEX `fk_Fornecido_Fornecedor1_idx` ON `ProdutoFornecedor`
(`Fornecedor_Id` ASC) VISIBLE;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

II. Anexo 2 – Script para povoamento do modelo físico

```
-- -----  
-- -----  
-- Universidade do Minho  
-- Mestrado Integrado em Engenharia Informática  
-- Unidade Curricular de Bases de Dados  
--  
-- Povoamento inicial da base de dados  
-- Dezembro/2020  
-- -----  
-- -----  
  
--  
-- Esquema: "LojaDesportiva"  
USE `SGBD` ;  
  
--  
-- Permissão para fazer operações de remoção de dados.  
SET SQL_SAFE_UPDATES = 0;  
  
--  
-- Povoamento da tabela "Cliente"  
INSERT INTO Cliente  
  (id, Nome, Localidade, CodigoPostal, Rua, DataNascimento, NIF, Email,  
  Observacao)  
  VALUES  
    (1, 'João da Costa e Campos',      'Aguada do Queixo', '9999-999',  'Rua  
das Adegas Felizes, 12, 1ª Cave',  '1983-12-31',123456789, 'jcc@acacia.pt',  
NULL),  
    (2, 'Josefina Vivida da Paz',      'Friso do Eixo',   '7799-777',  'Av dos  
Castros Reais, 122, 3º',      '1965-10-03',122133144, 'josefina@acacia.pt',  
NULL),  
    (3, 'Ana Santa do Carmo',          'Abre o Tacho',   '4534-541',  'Travessa do  
Jacob, 21',          '1990-08-12',876543298, 'saca@acacia.pt',  NULL),  
    (4, 'Jesualdo Peza-Mor',           'Vale dos Lençóis', '1245-641',  'Estrada  
do Sossego, Km10',      '1978-11-30',564352786, 'pezamor@acacia.pt', NULL),
```

```

        (5, 'Maria da Trindade Pascoal', 'Aguada do Queixo', '9999-999', 'Rua
das Adegas da Rua, 15, 10 Esq/T', '1982-02-05', 777666555,
'trindade@acacia.pt', NULL),
        (6, 'Florindo Teixeira Figueirinha', 'Veloandro', '5555-595', 'Autódromo
das Vagas, Garagem 123', '1970-05-22', 787676651, 'teixof@acacia.pt',
NULL),
        (7, 'Carminho Cunha Bastos', 'Vitalis do Sousa', '6532-243', 'Rua
do Mus-Vitalis, 56, r/c ', '1976-01-13', 543234111, 'cbastos@acacia.pt',
NULL)
    ;

--
-- DELETE FROM Cliente;
-- SELECT * FROM Cliente;

--
-- Povoamento da tabela "ClienteTelefones"
INSERT INTO TelefoneCliente
    (Telefone, Cliente_id)
VALUES
    ('678 546 234', 1),
    ('142 356 429', 2),
    ('998 765 420', 3),
    ('675 008 976', 4),
    ('776 879 332', 5),
    ('465 423 890', 6),
    ('866 543 420', 7),
    ('654 545 218', 6)
    ;

--
-- DELETE FROM ClienteTelefones;
-- SELECT * FROM ClienteTelefones;

--
-- Povoamento da tabela "Funcionario"
INSERT INTO Funcionario
    (Id, Nome)
VALUES
    (1, 'Miguelito Maciel'),
    (2, 'Alcides Venâncio'),
    (3, 'Dionísio Borges'),
    (4, 'Jorginho Soares'),
    (5, 'Ademar Andrade')
    ;

--
-- DELETE FROM Funcionario;

```

```

-- SELECT * FROM Funcionario;

--
-- Povoamento da tabela "Loja"
INSERT INTO Loja
  (Id, Nome, Tipo)
VALUES
  (1,'HQSportStoreInc','fisica'),
  (2,'OnlineSports' ,'online'),
  (3,'EveryDaySport'  ,'fisica')
;

--
-- DELETE FROM Loja;
-- SELECT * FROM Loja;

--
-- Povoamento da tabela "TelefoneLoja"
INSERT INTO TelefoneLoja
  (Texto, Loja_Id)
VALUES
  ('9600000001',1),
  ('9600000002',2),
  ('9600000003',3),
  ('9600000004',3)
;

--
-- DELETE FROM TelefoneLoja;
-- SELECT * FROM TelefoneLoja;

--
-- Povoamento da tabela "Armazem"
INSERT INTO Armazem
  (Id, Nome)
VALUES
  (1,'SportNStuffArmazem')
;

--
-- DELETE FROM Armazem;
-- SELECT * FROM Armazem;

--
-- Povoamento da tabela "Produto"
INSERT INTO Produto

```

```

(Id, PontoEncomenda, PrecoUnitario, Stock, UnidadeDeMedida,Descricao ,
PrecoMinimo,Armazen_Id)
VALUES
(1, 2, 42.00, 2, 'Un', 'Fato de treino', 30.00, 1),
(2, 3, 29.99, 4, 'Un', 'Botas', 20.00, 1),
(3, 4, 35.00, 8, 'Un', 'Sapatilhas', 25.00, 1),
(4, 2, 6.99, 3, 'Un', '5x Par de Meias', 4.00, 1),
(5, 3, 27.00, 1, 'Un', 'Calcas de Fato de treino', 21.00, 1),
(6, 2, 25.00, 5, 'Un', 'Hoodie', 19.00, 1),
(7, 5, 14.9, 7, 'Un', 'Casaco', 10.00, 1),
(8, 3, 10.00, 3, 'Un', 'T-shirt', 7.00, 1),
(9, 2, 4.99, 9, 'Un', 'Luvas', 4.00, 1)
;

--
-- DELETE FROM Produto;
-- SELECT * FROM Produto;

--
-- Povoamento da tabela "ProdutoLoja"
INSERT INTO ProdutoLoja
(Id, Loja_Id, Produto_Id, Stock)
VALUES
(1, 1, 1, 1),
(2, 2, 2, 1),
(3, 2, 5, 1),
(4, 3, 6, 1),
(5, 3, 2, 1)
;

--
-- DELETE FROM ProdutoLoja;
-- SELECT * FROM ProdutoLoja;

--
-- Povoamento da tabela "Compra"
INSERT INTO Compra
(Id, Nota, Comentario , Data , CustoTotal , Funcionario_Id,
Cliente_id,Loja_Id)
VALUES
(1, 5, NULL, '2020-11-11
13:23:44', 15.00, 1, 1, 1),
(2, 3, 'Demorou muito a chegar', '2020-11-12 14:33:24',
50.00, 2, 2, 2),
(3, NULL, NULL, '2020-11-10
14:31:21', 4.92, 3, 3, 3)
;

```

```

--
-- DELETE FROM Compra;
-- SELECT * FROM Compra;

--
-- Povoamento da tabela "ProdutoCompra"
INSERT INTO ProdutoCompra
    (Taxa, Quantidade, PrecoUnitario, Compra_Id, Produto_Id)
VALUES
    (0.00, 2, 4.00, 1, 9),
    (0.00, 1, 7.00, 1, 8),
    (0.00, 1, 30.00, 2, 1),
    (0.00, 1, 20.00, 2, 2),
    (0.24, 1, 4.92, 3, 4)
;

--
-- DELETE FROM ProdutoCompra;
-- SELECT * FROM ProdutoCompra;

--
-- Povoamento da tabela "Fornecedor"
INSERT INTO Fornecedor
    (Id, Nome)
VALUES
    (1, 'SportsStuff'),
    (2, 'Vestuario Desportivo'),
    (3, 'Calcado')
;

--
-- DELETE FROM Fornecedor;
-- SELECT * FROM Fornecedor;

--
-- Povoamento da tabela "ProdutoFornecedor"
INSERT INTO ProdutoFornecedor
    (Id, PrecoUnitario, Data, Quantidade, Produto_Id, Fornecedor_Id)
VALUES
    (1, 30.00, '2020-11-11 13:23:44', 2, 1, 2),
    (2, 20.00, '2020-11-11 13:23:44', 4, 2, 3),
    (3, 25.00, '2020-11-11 13:23:44', 8, 3, 3),
    (4, 4.00, '2020-11-11 13:23:44', 3, 4, 2),
    (5, 21.00, '2020-11-11 13:23:44', 1, 5, 1),
    (6, 25.00, '2020-11-11 13:23:44', 5, 6, 2),
    (7, 10.00, '2020-11-11 13:23:44', 7, 7, 2),
    (8, 7.00, '2020-11-11 13:23:44', 3, 8, 2),

```

```
        (9, 4.00, '2020-11-11 13:23:44',    4,    9,    1),
        (10,4.00, '2020-11-11 13:23:44',    5,    9,    2)
    ;
--
-- DELETE FROM ProdutoFornecedor;
-- SELECT * FROM ProdutoFornecedor;

-- -----
-- <fim>
-- -----
```


III. Anexo 3 – Script para execução de queries

```
USE `SGBD`;

-- Query 1: Qual o total de vendas do Produto com Id = 1?
SELECT SUM(Quantidade) AS Total FROM ProdutoCompra WHERE Produto_Id = 1;

-- Query 2: Qual o catalogo de produtos da Loja com Id = 3?
SELECT DISTINCT(Descricao) AS Catalogo FROM ProdutoLoja
      JOIN Produto ON Loja_Id = 3;

-- Query 3: Quais os nomes das lojas registradas no sistema?
SELECT Nome AS Nomes FROM Loja;

-- Query 4: Qual o total de faturação durante o mês de março de 2020?
SELECT SUM(CustoTotal) AS Faturacao FROM Compra
      WHERE `Data` BETWEEN '2020-03-01 00:00:00' AND '2020-03-31 23:59:59';

-- Query 5: Quais são o top10 melhores Clientes do Sistema, dado pelo nome e
total de compras associado
SELECT Nome ,COUNT(Cliente_Id) AS TotalCompras FROM Cliente
      JOIN Compra ON Compra.Cliente_Id = Cliente.Id
      GROUP BY Cliente.Id
      ORDER BY TotalCompras DESC
      LIMIT 10;

-- Query 6: Compras realizadas pelos clientes com id = 1 2 e 3 durante o mes
de Marco de 2020
SELECT Cliente.Nome FROM Cliente
      JOIN Compra ON Cliente.id = Compra.Cliente_Id
      WHERE Cliente.id >=1 AND Cliente.id<=3 AND year(Compra.`Data`)=2020 AND
month(Compra.`Data`)=03;

-- Query 7: Quais os valores faturados e produtos que foram vendidos aos
clientes com mais de 40 anos
SELECT DISTINCT P.Descricao, PC.Quantidade * PC.PrecoUnitario AS TOTAL FROM
Produto AS P
      JOIN ProdutoCompra AS PC ON P.id=PC.Produto_Id
      JOIN Compra AS C ON PC.Compra_Id = C.Id
```

```

JOIN Cliente AS Cl ON C.Cliente_id = Cl.Id
WHERE year(NOW())-year(Cl.DataNascimento)>=40;

-- Query 8: Quais os tipos 5 de produtos mais vendidos durante a semana 15 de
2020
SELECT DISTINCT P.Descricao, SUM(PC.Quantidade) AS Soma FROM Produto AS P
JOIN ProdutoCompra AS PC on P.Id = PC.Produto_Id
JOIN Compra AS C on PC.Compra_Id = C.Id
    WHERE week(C.`Data`) =15 AND year(C.`Data`)=2020
    group by P.Id
    order by Soma desc
LIMIT 5;

-- Query 9: Quais os produtos nunca comprados
SELECT P.Id, P.Descricao FROM Produto AS P
LEFT JOIN ProdutoCompra as PC ON P.Id = PC.Produto_Id
WHERE PC.Compra_Id is null;

-- Query 10: Quais os clientes que nunca compraram nada
SELECT Cl.id, Cl.Nome FROM Cliente AS Cl
WHERE Cl.id NOT IN (SELECT Cl.id FROM Cliente AS Cl
    JOIN Compra AS C ON Cl.id=C.Cliente_id
    JOIN ProdutoCompra AS PC ON C.Id = PC.Produto_Id);

-- Query 11: Qual o produto mais comprado no sistema, a quantidade comprada e
o total faturado
SELECT P.Descricao, PC.Quantidade, SUM(PC.Quantidade*PC.PrecoUnitario) AS
Valor FROM Produto AS P
JOIN ProdutoCompra AS PC ON P.Id = PC.Produto_Id
JOIN Compra as C ON PC.Compra_id = C.Id
GROUP BY P.Descricao;

-- Procedure 1: Qual o produto mais comprado no sistema, a quantidade
comprada e o total faturado, No ultimo mes
delimiter $$
    CREATE PROCEDURE produtosPopularesUltimosMeses(n Int)
    begin
        SELECT
            P.Descricao,
            PC.Quantidade,
            SUM(PC.Quantidade*PC.PrecoUnitario) AS Valor FROM Produto AS P
            JOIN ProdutoCompra AS PC ON P.Id = PC.Produto_Id
            JOIN Compra as C ON PC.Compra_id = C.Id
            WHERE month(NOW())- month(C.`Data`) <=n
        GROUP BY P.Descricao

```

```

        order by PC.Quantidade desc;
    end$$
call produtosPopularesUltimosMeses(2);

-- Vista 1: Disponibilizar uma lista com as datas de aniversario e atuais dos
clientes atuais do sistema

CREATE VIEW vwListaAniversariosIdades AS
    SELECT Cliente.DataNascimento, datediff(NOW(),(Cliente.DataNascimento))
div 365 AS Idades FROM Cliente;

SELECT * FROM vwListaAniversariosIdades;

-- Vista 2: Disponibilizar uma lista com o numero total das compras por
localidade dos clientes. A lista deverá ser apresentada por ordem decrescente
da numero total de compras

CREATE VIEW vwListaComprasPorLocalidade AS
    SELECT Cl.localidade, COUNT(C.Cliente_Id) AS `Total de Compras` FROM
Cliente AS Cl
        JOIN Compra AS C ON C.Cliente_Id = Cl.Id
        GROUP BY Cl.Id
        ORDER BY `Total de Compras` desc;

SELECT * FROM vwListaComprasPorLocalidade;

CREATE INDEX idx_Data ON Compra (`Data`);

SELECT table_schema `loja`
    , (SUM(data_length + index_length)) "Database Size in Bytes"
FROM information_schema.TABLES
GROUP BY table_schema;

```