­­

**BD**

**Universidade do Minho**

Mestrado Integrado em Engenharia Informática

**Unidade Curricular de**

**Bases de Dados**

Ano Letivo de 2018/2019

**Mercado de compra e vendas online**

**Luís Capa A81960**

**Moisés Antunes A82263**

**Pedro Capa A83170**

**Tiago Pinheiro A82491**

novembro, 2018

|  |  |
| --- | --- |
| Data de Recepção |  |
| Responsável |  |
| Avaliação |  |
| Observações |  |

**Resumo**

<<O resumo tem como objectivo descrever de forma sucinta o trabalho realizado. Deverá conter uma pequena introdução, seguida por uma breve descrição do trabalho realizado e terminando com uma indicação sumária do seu estado final. Não deverá exceder as 400 palavras.>>

**Área de Aplicação:** <<Identificação da Área de trabalho. Por exemplo: Desenho e arquitetura de Sistemas de Bases de Dados.>>

**Palavras-Chave:** <<Conjunto de palavras-chave que permitirão referenciar domínios de conhecimento, tecnologias, estratégias, etc., direta ou indiretamente referidos no relatório. Por exemplo: Bases de Dados Relacionais, Gestão de Índices, JAVA, Protocolos de Comunicação.>>

**Acrónimos:** Unidade Curricular (UC), Base de Dados (BD), Gigabyte (GB)

**Índice**

[1. Introdução 1](#_Toc535645383)

[1.1. Contextualização 1](#_Toc535645384)

[1.2. Apresentação do Caso de Estudo 1](#_Toc535645385)

[1.3. Motivação e Objectivos 1](#_Toc535645386)

[1.4. Estrutura do Relatório 1](#_Toc535645387)

[2. Sugestões para Escrita do Relatório 2](#_Toc535645388)

[2.1. Sugestões Gerais 2](#_Toc535645389)

[2.2. Termos Estrangeiros 2](#_Toc535645390)

[2.3. Tabelas e Figuras 2](#_Toc535645391)

[2.4. Siglas e Acrónimos 3](#_Toc535645392)

[2.5. Referências Bibliográficas 3](#_Toc535645393)

[2.6. Tipo de Ficheiro 3](#_Toc535645394)

[3. Conclusões e Trabalho Futuro 4](#_Toc535645395)

[Bibliografia 5](#_Toc535645396)

[Referências WWW 6](#_Toc535645397)

[Lista de Siglas e Acrónimos 7](#_Toc535645398)

**Anexos**

[I. Anexo 1 9](#_Toc535644882)

**1. Introdução**

O aumento do uso da Internet tornou possível realizar algumas atividades sem sair de casa, entre as quais, comprar e vender. Num site de compras online estão disponíveis todo o tipo de produtos, desde material escolar até roupa existindo uma maior variedade de produtos, uma vez que alguns produtos são difíceis de encontrar em lojas físicas. Como não há necessidade de ir à loja o gasto em tempo e em dinheiro diminui, estando esta disponível a qualquer altura. Pelo contrário há um aumento da privacidade, pois pode-se comprar qualquer produto sem ser notado. Por estes motivos este tipo de mercado tem tido bastante sucesso.

**1.1 Contextualização**

Foi-nos proposto criar uma base de dados na *UC* de *Base de Dados (BD)*, com a finalidade de avaliar a nossa capacidade na construção de esquemas, concetual e lógico, e na construção de *queries*. Cada grupo escolhia o tema do seu trabalho. A escolha foi “Mercado de Compra e Vendas *Online*”, pois alguns membros usam com alguma frequência este tipo de mercado.

**1.2 Apresentação do caso de Estudo**

Neste trabalho focamo-nos no site online Amazon.com, uma vez que é a empresa online com maior sucesso da atualidade. Sendo um dos primeiros sites deste género e devido ao sucesso todo que obteve o dono da Amazon é neste momento o homem mais rico do mundo e foi a segunda empresa do mundo a atingir o valor de 1 trilião de dólares.

O site online Amazon é o maior sítio da internet no que se refere a compra e venda de produtos. Este vende artigos em 1ª mão de grandes marcas como a Samsung e a LG e artigos em 2ª mão que qualquer utilizador que esteja registado no Amazon pode fazer. O Amazon, como qualquer site online, podem ser pesquisados artigos em específico, como o Xiaomi Redmi 5 Plus, ou procurar um tipo de produtos como smartphones.

**1.3 Motivação e objetivos**

No nosso percurso académico não tivemos contacto direto no que se refere a construções de bases de dados. Esta *UC* foi projetada para praticar todos os aspetos relevantes de um sistema de bases de dados, quer estes sejam relacionais ou não relacionais. Para isso foi proposto a realização de um trabalho com a finalidade de melhorar as nossas competências na construção e gestão de um banco de dados.

Como em praticamente todos os aspetos da informática é necessário uma base de dados a aprendizagem da gestão da mesma torna-se importante para o nosso futuro, uma vez que “abre uma porta” naquilo que poderemos vir a fazer no futuro.

**2. Levantamento e análise de requisitos**

**2.1 Métodos de levantamento e análise de requisitos adotados**

No processo de levantamento e análise de requisitos foi utilizada uma abordagem centralizada, ou seja, foram seguidos os seguintes passos:

* **Coleta de requisitos**: Recolha de requisitos, sobre o sistema, com potenciais clientes do site;
* **Reunir**: Agrupar os vários requisitos dos diferentes utilizadores em uma única lista de requisitos
* **Verificação de requisitos**: Estudo dos requisitos levantados de forma a garantir que não haja inconsistências, ou conflitos, entre diferentes requisitos;
* **Validação de requisitos**: Garantir que todas as principais funcionalidades foram cumpridas com precisão e que o sistema seja implementável;

**2.2 Requisitos Levantados**

**2.2.1 Requisitos de Descrição**

Para que um usuário pudesse utilizar este site de compra e vendas era pedido as seguintes informações sobre este, tais como o nif, que era o seu identificador, a password, estas duas eram necessárias se o utilizador quisesse autenticar, a morada, visto que as compras eram entregues ao domicílio, a sua data de nascimento, uma lista de contactos, email e telefone, e ainda uma lista de métodos de pagamento, ou seja, os métodos que dispõe para carregar a conta, um utilizador também tem um saldo.

Os produtos que são colocados à venda tem um identificador, uma designação, descrição, preço, quantidade e têm de estar associados a um utilizador, o que o colocou para vender.

Uma compra é referente a um só produto, no entanto, e possível escolher a quantidade que se pretende comprar deste produto. Logo, a compra tem uma quantidade, o preço total da compra e está associada a um carrinho.

O carrinho é uma lista de compras, na qual regista-se a data das compras e o método de transporte que o utilizador escolhe para entregar os produtos.

Para efetuar operações sobre o saldo o utilizador tem diversos métodos de pagamento disponíveis. Também é fornecido algumas formas de entrega, ou métodos de transporte, que está associado um custo e um tempo estimado de entrega.

**2.2.2 Requisitos de exploração**

Requisitos do utilizador

* Criar uma conta na aplicação, colocando alguns dados para se registar;
* Ver/Comprar produtos que estão disponíveis no mercado;
* Filtrar os produtos que pretende ver/comprar do mercado;
* Colocar à venda um artigo;
* Aceder a todas as compras realizadas;
* Adicionar/Alterar/Aceder informações pessoais;
* Ver informações detalhadas sobre cada forma de pagamento ou transporte;
* Carregar/Levantar dinheiro para/da conta;

Requisitos do administrador

* Aceder à lista de todos os produtos no sistema;
* Aceder a informações pessoais de todos os utilizadores;
* Aceder a todas as compras no sistema;
* Adicionar/Remover métodos de pagamento ou transporte;
* Ver a faturação total;
* Ver a faturação de cada cliente;
* Ver estatísticas do mercado e dos utilizadores;

**2.2.3 Requisitos de Controlo**

Para que a dados se mantivessem privados e a base de dados se mantivesse confiável foi decidido que cada um dos tipos de utilizador teria os seguintes privilégios.

Requisitos do utilizador

* + Comprar produtos que estão disponíveis no mercado;
  + Adicionar/Alterar informações pessoais;
  + Colocar um artigo à venda;
  + Carregar/Levantar dinheiro para/da conta;
  + Criar uma conta na aplicação, colocando alguns dados para se registar;

Requisitos do administrador:

* + Adicionar/Remover métodos de pagamento ou transporte;

**2.3 Análise geral dos requisitos**

Após o levantamento dos requisitos foi organizado uma entrevista com o cliente de forma a perceber se estes estavam de acordo com as suas ideias. Segundo os requisitos levantados, chegou-se à conclusão que atendem às funcionalidades básicas para o utilizador usar a aplicação, ao mesmo tempo restringindo os seus direitos.

**3. Modelo Conceptual**

**3.1 Apresentação da abordagem de modelação realizada**

O primeiro passo na construção do modelo concetual foi definir as principais entidades, para isso foram analisados os requisitos do sistema e identificar todos os nomes. A seguir foram identificadas as entidades e associados os diferentes atributos a cada entidade, e o seu tipo, por exemplo se eram identificadores da entidade, atributos multivalorado, atributo derivado ou um atributo composto. A seguir foi procurar as principais relações entre as várias entidades e o tipo de relações e a multiplicidade das relações. Por fim, foi verificado se havia redundância no modelo e se este estava de acordo com os requisitos.

**3.2 Identificação e caracterização das entidades**

* Utilizador
* Métodos de Pagamento
* Produto
* Compra
* Carrinho
* Método de Transporte

Como seria de esperar, uma das entidades é o utilizador. Este é o responsável pelas operações de compra ou venda no site. Este dispõe de diferentes métodos de pagamentos (que são por si só uma entidade) como transferência bancária, cartão de crédito e outros mais. Como foi mencionado acima, o utilizador pode vender ou comprar produtos no site, sendo que os produtos, como entidades, devem ser listados e caraterizados pelo seu vendedor. Várias compras individuais de produtos formam um carrinho de compras, ao qual os produtos serão adicionados após as compras, que serão oficializadas dada a confirmação do carrinho final. Os produtos comprados serão então transportados até ao seu comprador pelo método mais conveniente.

**3.3 Identificação e caracterização dos relacionamentos**

* + Utilizador – Métodos de Pagamento
  + Utilizador – Produto (Venda Apenas)
  + Produto – Compra
  + Compra – Carrinho
  + Utilizador – Carrinho
  + Carrinho – Método de Transporte

O método de pagamento é escolhido pelo utilizador que dispõe de várias opções de como quer pagar o(s) produto(s) que vai comprar. O utilizador deve efetuar a compra adquirindo os produtos que pretende, sendo que a compra individual de cada produto será automaticamente adicionada ao carrinho de compras do utilizador, no qual estes aguardam confirmação do comprador. O carrinho é depois atribuído a um método de transporte, através do qual chegará ao seu comprador. Note-se que existe uma relação direta entre utilizador e produto, mas neste caso o utilizador é necessariamente o vendedor, já que o comprador se relaciona com o produto através das entidades que lhe permitem fazer a compra.

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

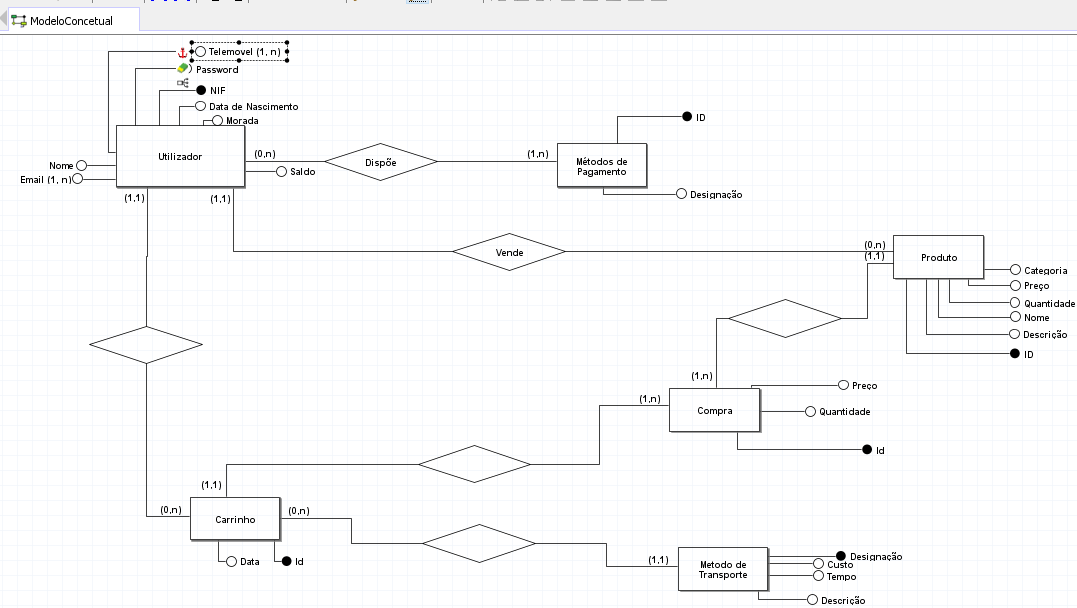
* Utilizador: NIF, Data de Nascimento, Morada, Nome, Password, Telemóvel, Email e Saldo.
* Métodos de Pagamento: ID e Designação.
* Produto: ID, Categoria, Preço, Quantidade, Nome e Descrição.
* Compra: ID, Preço e Quantidade.
* Carrinho: ID e Data.
* Método de Transporte: Designação, Custo, Tempo e Descrição.

O utilizador necessita de fornecer dados pessoais para se cadastrar no site e o utilizar. O NIF é necessário para realizar transferências de dinheiro (que será adicionado ou retirado do saldo), a data de nascimento para determinar a idade do utilizador (por exemplo, para proibir acesso a menores), o nome e a password para garantir apenas ao utilizador acesso à sua conta pessoal, morada para assuntos das entregas e contatos caso necessário (sendo que o nº de telemóvel tem obrigatoriamente nove algarismos). Cada método de pagamento tem um nº de identificação, assim como cada produto, compra e carrinho. Os produtos são organizados por categoria e é possível atribuir-lhes, para além do nome e preço base, a quantidade a vender e uma breve descrição. A compra detalha, para além da quantidade de produtos comprados, o preço a que o produto foi vendido, que pode ser diferente do preço base em caso de ter havido alteração do preço devido a algum desconto, negociação entre os intervenientes ou outro motivo. O carrinho mostra a data da compra e o método de transporte é identificado pelo nome, tendo também portes de envio, tempo de entrega e descrição do método.

**3.5 Detalhe ou generalização de entidades**

É possível ver através das alíneas anteriores que houve destas situações nas entidades. Um caso de generalização é o fato de o comprador e o vendedor serem ambos instâncias da mesma entidade, neste caso a entidade “Utilizador”. Um caso de detalhe está nas entidades “Produto” e “Compra”, nas quais temos diferentes preços conforme o preço listado e aquele que foi de fato pago.

**3.6 Apresentação e explicação do diagrama ER**

****

Estas entidades e atributos foram os escolhidos, como referido nas secções anteriores, nomeadamente as secções 3.2 e 3.4, sobretudo devendo se ter em conta que os atributos referentes aos contactos do utilizador são multivalorados e que através do esquema é fácil perceber quais dos atributos são chaves primárias. Os relacionamentos criados e os seus tipos são os explicados na secção 3.3, sendo que o esquema concetual ajuda a perceber que a relação “Utilizador-Produto” é apenas em caso de venda e não de compra, ou seja, um comprador relaciona-se com o produto que compra através das entidades “Compra” e “Carrinho”.

**3.7 Validação do modelo de dados com o utilizador**

O esquema concetual inclui o diagrama ER e a documentação necessária para o descrever. Por vezes, o esquema não corresponde ao pensamento inicial, portanto é essencial rever e alterar o esquema conforme seja preciso. Isso implica por vezes voltar atrás e alterar entidades, relacionamentos e atributos até que o esquema esteja o mais próximo possível do que pretendemos que seja o resultado final.

Um exemplo disso foi a criação da entidade “Fatura”, que agora não se encontra no esquema, que continha informações sobre as compras e o carrinho, tais como o preço pago e a data da compra. Tal entidade foi eventualmente retirada, isto porque se considerou que as restantes entidades poderiam assumir as suas funções, tornando o modelo mais compacto e eficaz.

**4. Modelação Lógica**

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

O modelo de dados lógico foi construído a partir do modelo conceptual feito anteriormente, usando-o como um esquema base. Depois para efetuar a validação é preciso garantir que o modelo é estruturalmente correto e suporta todos os requerimentos desejados.

Para atingir o sucesso na construção e validação de um modelo de dados lógico, é preciso que todos os seguintes aspetos sejam cumpridos com sucesso:

- Derivar as relações para o modelo lógico – partindo do modelo concetual tem de ser efetuada uma correta transição das relações, por outras palavras, as relações que entidades que existam no concetual tem de ter essa mesma relação no modelo lógico.

- Os restantes aspetos são enunciados e sucintamente explicados nos pontos 4.3 a 4.7 deste relatório.

 **4.2 Desenho do modelo lógico**

**4.3 Validação do modelo através da normalização**

O modelo é válido através da normalização se respeitar as três primeiras regras da forma normal.

Analisando o modelo, verificamos que, no modelo, as tabelas existentes não têm valores repetidos e os atributos são atómicos possuindo no máximo, um valor. Assim, concluímos que o modelo respeita a Primeira Forma Normal.

Garantindo a verificação da primeira forma normal, podemos prosseguir para a verificação da segunda. Através da observação das tabelas, conseguimos concluir que todos os atributos normais, os que não possuem chave, são unicamente dependentes da chave primária dessa tabela. Deste modo, verificamos a Segunda Forma Normal.

Por último, depois de garantidas as duas primeiras, falta verificar a terceira forma normal para o modelo ser válido. Como em todas as tabelas do modelo os atributos são independentes entre si e dependentes da respetiva chave primária, podemos afirmar que o modelo está normalizado até à Terceira Forma Normal.

**4.4 Validação do modelo com interrogações do utilizador**

Através de uma conversa com o utilizador do sistema, obtivemos a validação do nosso modelo lógico.

Durante a conversa, o utilizador fez inúmeras questão para verificar o modelo, como o nosso modelo conseguiu responder as cenas interrogações, obtivemos a validação do mesmo.

**4.5 Validação do modelo com as transações estabelecidas**

Para efetuar a validação do modelo com as transações estabelecidas é necessário tentar resolver as várias operações manualmente.

Depois de resolvidas todas as transações usando este método podemos concluir que as relações presentes no nosso esquema lógico e o nosso esquema suportam as transações pedidas.

Assim, o modelo lógico é válido perante as transações estabelecidas.

**4.6 Revisão do modelo lógico com o utilizador**

Depois nosso modelo estar acabado e devidamente documento foi exposto a um utilizador que iria avaliar, para garantir que o modelo seria válido e ideal para os requerimentos desejados.

Depois de o modelo lógico ter sido revisto pelo utilizador, ele comunicou que o nosso modelo ia de encontro ao modelo pedido, ou seja, respondendo muito bem a todos os requerimentos pedidos.

**5. Implementação Física**

**5.1 Seleção do sistema de gestão de bases de dados**

Para gerir a *BD* foi usado um sistema relacional que permite manter a segurança, integridade e consistência de dados. Este tipo de modelo suporta relações entre diferentes entidades em tabelas através de chaves estrangeiras. As chaves primárias permitem normalizar os dados evitando redundância de dados das tabelas. Por estes motivos é possível criar, atualizar, consultar e gerir a *BD,* de uma forma mais simples*.*

O sistema de gestão de base de dados (*SGBD*) utilizado foi o proposto pelos docentes da *UC,* o *Mysql*. Este sistema usa algoritmos complexos que suportam a concorrência no acesso à BD, enquanto mantêm a sua integridade.

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

No desenho de uma base de dados após a criação do modelo lógico do sistema, este é “traduzido” para um esquema físico.

Os esquemas têm “traduções” entre si, por exemplo, as entidades no modelo físico são referidas como tabelas e os atributos de cada entidade são colunas. A maior diferença entre os esquemas é que o modelo físico particulariza o tipo de dados, por exemplo, específica que a chave primária dos utilizadores, o *NIF*, é um inteiro, ou um que a data é do tipo date. Sendo esta a maior diferença a tradução não foi um passo complicado na construção da *BD*.

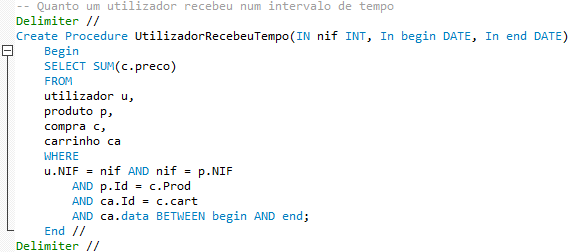
**5.3 Tradução das interrogações do utilizador para SQL (alguns exemplos)**

Foram muitas as exigências do utilizador quanto às funcionalidades do programa. Neste subcapítulo iremos mostrar apenas algumas delas.

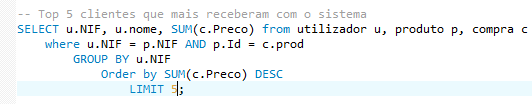
 Para verificar quais são os utilizadores do sistema a *query* tem um grau de dificuldade baixo. É visto na tabela todas as informações de todos os utilizadores.

Uma informação mais requisitada é a verificação de todos os produtos disponíveis para venda, uma vez que esta informação é consultada sempre que um utilizador acede ao site. Os produtos estão disponíveis no caso de a quantidade ser maior a 0.

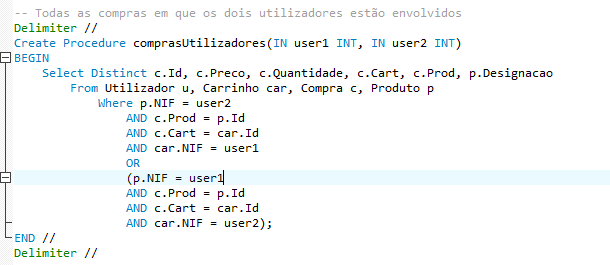


Para verificar quanto um utilizador recebeu num intervalo de tempo usa-se um *procedure* que recebe como argumento o *NIF* do utilizador que se pretende verificar, bem como duas datas. No procedimento filtram-se os produtos que pertencem ao utilizador. Verificam-se quais os produtos estão na tabela de compras dentro do intervalo de tempo e soma-se o valor total.

Para verificar quais são os 5 utilizadores que mais faturaram com o sistema agrupa-se os produtos aos utilizadores. Somam-se os preço, ordena-se por ordem decrescente e retira-se os 5 que mais receberam.

****

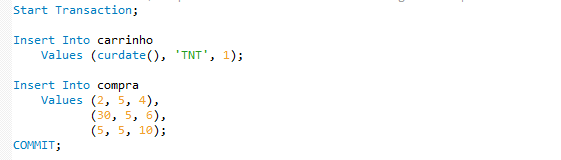
O objetivo da seguinte *query* era ver todas as compras nas quais dois utilizadores estavam simultaneamente envolvidos. Para começar foram filtrados todos os produtos que o utilizador 2 tinha colocado à venda. Depois, foram unidas as tabelas que continham compras referentes aos produtos do utilizador 2. Também foram filtrados todos os carrinhos no qual o utilizador 1 estava envolvido. Por fim, foram unidas as duas tabelas resultantes pelo atributo “Cart” da compra e o “Id” do carrinho. Foi utilizado o operador OR, mas desta vez os dois utilizadores tinham os papéis trocados.



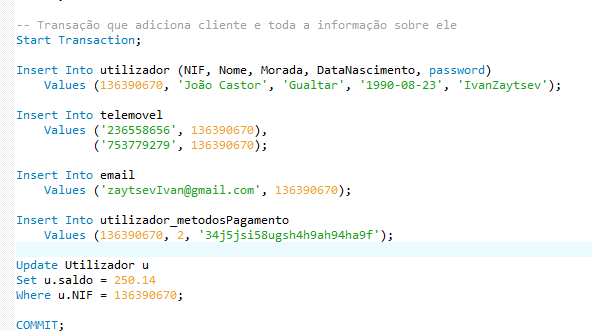
**5.4 Tradução das transações estabelecidas para SQL (alguns exemplos)**

Para garantir maior consistência na *BD*, foram criadas algumas transações. Neste capítulo são dados alguns exemplos de transações, nas quais foi considerado que as interrogações estarem relacionadas.

A transação abaixo é, basicamente, o registo de uma operação de compra. Primeiro, adiciona-se o carrinho e depois, são adicionadas as compras desse carrinho.



Nesta transação contêm o processo de criação de um utilizador para a *BD.* Para começar foi inserido a informação principal do cliente, a seguir, foram adicionados os contactos deste, tal como os métodos de pagamento. Por fim, foi adicionada o saldo do cliente.



**5.5 Escolha, definição e caracterização de índices em SQL (alguns exemplos)**

Os índices são fundamentais em *BD*, visto que permitem uma maior facilidade na gestão da mesma.

Aceder à informação de uma linha em que a tabela tenha índices é mais rápido pois acede-se diretamente à informação no caso da chave existir na tabela e sabe-se que se retira sempre o que se pretende pois não existem duas linhas com chaves iguais. Estes podem ser comparados com os *Map,* que por exemplo, são usados em programação orientado aos objetos. As tabelas já vêm por *default* ordenadas pela chave primária e como estas são as mais usadas em *joins* e no acesso ás tabelas, então estas tarefas são mais eficientes.

Na *BD* do projeto são usados índices, por exemplo, para identificar o utilizador. Sabendo que os *NIF* são únicos, é garantido que a informação que é fornecida sobre os seus dados está correta. Para o caso em que os ids apenas servem para relações, por exemplo, para identificar que produtos fazem parte de uma compra, é usado *auto incremente.* Este método está presente nos índices do produto, compras e carrinho.

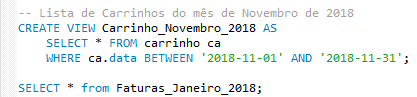
**5.6 Estimativa do espaço em disco da base de dados e taxa de crescimento anual**

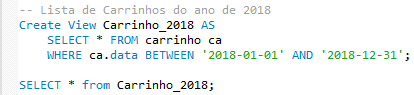
Este é um passo importante na criação da base de dados, pois pode ser necessário adquirir novo *hardware* para armazenar a *BD*, no presente bem como no futuro dependendo da taxa de crescimento da *BD*. Caso esta etapa não seja bem feita, o espaço ocupado do disco atinja uma percentagem muito elevada e não seja feito nada para ampliar a capacidade de armazenamento da *BD*, este tornar-se há muito lento bem como ser impossível adicionar mais informação á *BD*.

Como a *BD* inicialmente apenas tem apenas as tabelas e algumas informações *default*, como por exemplo, os métodos de pagamento ou os formas de transporte disponível, então o tamanho da base de dados será muito pequeno. No entanto é esperado um crescimento exponencial dos utilizadores do *site* nos dois primeiros anos bem como o número de produtos colocados postos à venda, devido à promoção do *site* e a partir desse momento seja linear. Por estes motivos espera-se que o tamanho da base de dados cresça em 50 *GB* por ano nos dois primeiros anos e 25 *GB* por ano nos anos seguintes.

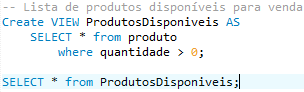
**5.7 Definição e caracterização das vistas de utilização em SQL (alguns exemplos)**

As vistas em *sql* podem tornar a *BD* mais eficiente. No caso de um acesso a uma tabela com algumas restrições ser pouco eficiente e ser utilizado com alguma frequência podem ser criadas vistas com a finalidade de um acesso à informação mais eficiente. No caso deste projeto foram pedidas informações mensais e anuais acerca dos carrinhos realizados.





Para os produtos como a tabela seria muito extensa então verificar quais são os produtos disponíveis para a venda iria ser uma operação pesada então a view dos produtos será atualizada frequentemente.



**5.8 Definição e caracterização dos mecanismos de segurança em SQL (alguns exemplos)**

A *BD* representa uma parte essencial de uma aplicação, portanto é fundamental que esta esteja protegida. Falhas de segurança na *BD* podem afetar outras partes do sistema.

Há várias formas de aumentar a segurança como:

* Limitar a informação que cada utilizador possa aceder;
* Backup

As *passwords* e os *NIF*, apenas podem ser acedidas pelos utilizadores das contas. As restantes informações são públicas e podem ser acedidas por todos os utilizadores.

Uma *BD* na qual sejam feitos regularmente Backups tem um risco menor de perder informação. Caso haja uma falha no disco a informação poderá nunca mais ser recuperada. Se houver um Backup atualizado os dados não serão perdidos.

**5.9 Revisão do sistema implementado com o utilizador**

Após o sistema ser implementado, é necessário revê-lo para assegurar o seu funcionamento apropriado. Para uma base de dados trabalhar como deve ser, o essencial deve ser ter em conta que o seu objetivo principal é guardar e aceder a dados eficientemente. A sua eficiência pode ser medida ao observar vários fatores, tais como a taxa de transferência, o tempo de resposta e o espaço ocupado na memória. Idealmente, pretende-se que todos esses três valores sejam o mais baixos possível, mas tal é difícil de concretizar. Geralmente, de maneira a minimizar um destes fatores, outros serão sacrificados, mas o ideal será procurar um equilíbrio entre os três. Quando este equilíbrio é encontrado, surgem vantagens tanto a nível da máquina, como melhor desempenho, menos hardware e manutenção deste, como também para as pessoas, já que a eficiência do programa satisfaz tanto o(s) seu(s) criador(es) como o cliente.

**6. Conclusões e Trabalho Futuro**

**7. Referências Bibliográficas (em formato Harvard)**  
  
**Anexos**