

# Online Marketplace

Plataforma para compra e venda de produtos

**Mestrado Integrado em Engenharia Informática e Computação 2020/2021**

Unidade Curricular: Bases de Dados

## Grupo 704

Catarina Pires	up201907925@fe.up.pt
Miguel Amorim	up201907756@fe.up.pt
Pedro Gonalo Correia	up201905348@fe.up.pt

# Índice

Contexto.....	3
Diagrama UML .....	5
Diagrama UML Revisto.....	6
Esquema Relacional .....	7
Análise de dependências funcionais e formas normais.....	8
Restrições.....	10
Interrogações .....	14
Gatilhos .....	16

## Contexto

Pretende-se criar um serviço de comércio online para compra e venda de produtos entre os seus utilizadores.

Este serviço contará com duas possibilidades alternativas de registo de novos utilizadores: enquanto individual ou enquanto empresa. Em ambos os casos, pretende-se armazenar o nome de utilizador, o e-mail e a data de registo na plataforma. Adicionalmente, caso esteja a ser feito o registo de uma conta pessoal, será necessário saber o nome próprio e a data de nascimento, sendo apenas permitido o registo a maiores de 18 anos. Por outro lado, se o registo se der para uma empresa, será conveniente guardar o nome dessa empresa, a morada onde está sediada e o domínio do seu website.

Qualquer utilizador poderá inscrever-se como comprador e/ou vendedor na plataforma, sendo obrigado a escolher pelo menos uma dessas duas opções aquando do seu registo. Assim, a interface da plataforma estará ajustada ao papel que o utilizador pretende ter, mostrando apenas as funcionalidades que lhe são relevantes. No caso de o utilizador pretender inscrever-se como vendedor, deve ainda fornecer o seu contacto telefónico, para facilitar o contacto por parte de quem tiver interesse nas suas ofertas, e ficará associado a uma classificação média, cujo cálculo será explicado mais adiante, que permite dar a potenciais compradores uma perceção do mesmo.

A principal funcionalidade relevante a um vendedor é a criação de ofertas para a venda de produtos. Uma oferta possui um título, a data da sua publicação na plataforma, o nome do produto que está a ser vendido e se este é ou não usado, bem como uma breve descrição com aspetos que o vendedor considere relevante. Inclui também o *stock* e preço por unidade do produto, se este preço é negociável e a morada onde o produto poderá ser levantado após a compra. A oferta poderá incluir ainda fotografias do produto. Para tal, o vendedor faz *upload* dessas imagens para o servidor da plataforma, devendo, portanto, ser armazenada a localização (*path*) dos respetivos ficheiros. Como é feito *upload* de cada vez que se inclui uma imagem, é impossível que o mesmo ficheiro esteja presente em duas ofertas distintas. Uma oferta tem de pertencer a uma ou mais categorias, que a ajudam a ser descoberta por potenciais interessados ao navegar na plataforma.

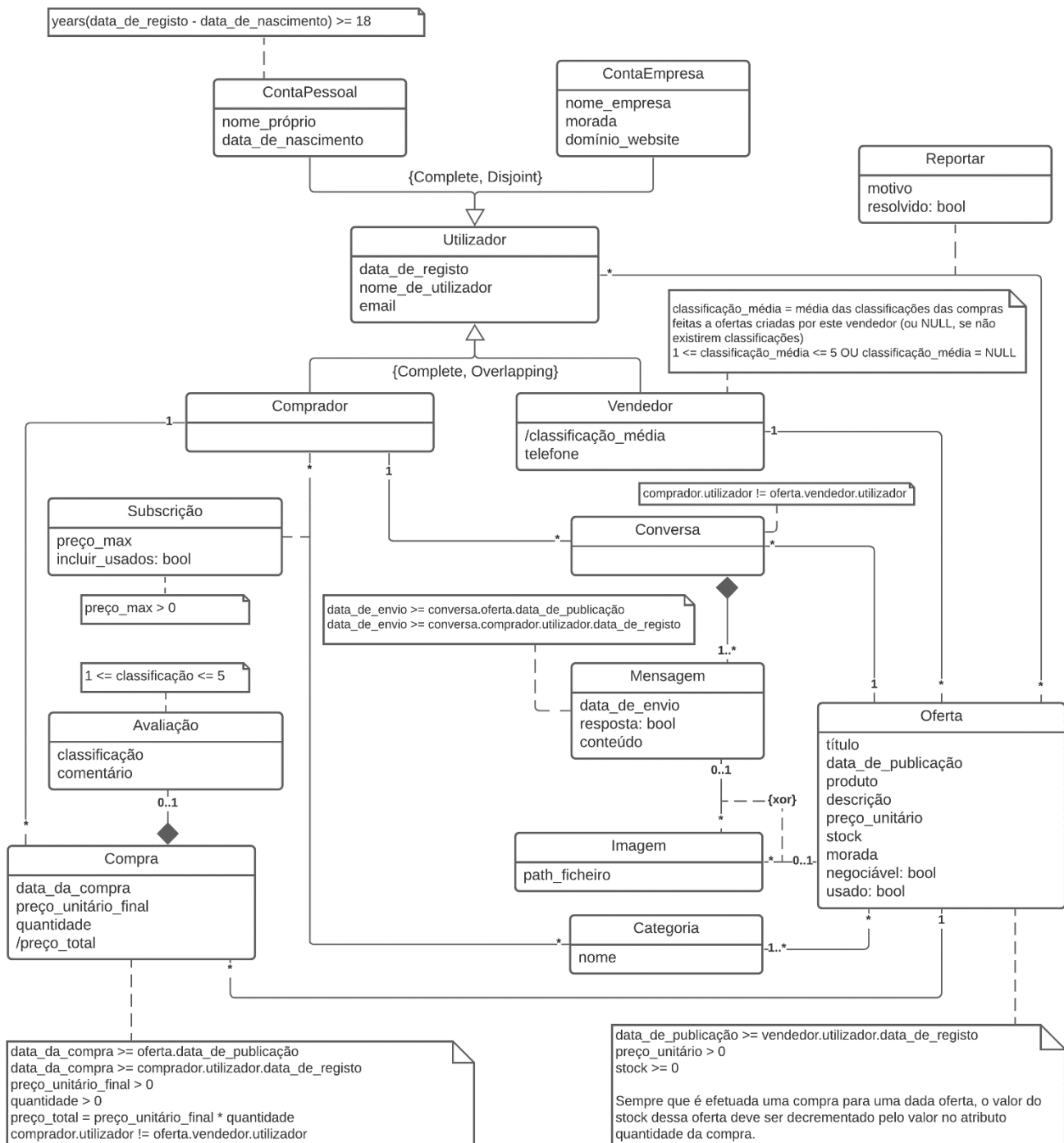
Qualquer utilizador terá a opção de reportar ofertas, por exemplo, pela presença de conteúdo impróprio ou por serem burlas. Ao reportar uma oferta, é preciso explicitar o motivo num breve texto. Para garantir que todos os *reports* são analisados, é necessário saber quais ainda estão por resolver, sendo que após a análise de um *report* este será marcado como resolvido.

A principal funcionalidade relevante a um comprador é a compra de produtos de uma dada oferta. Dessa compra deve ficar registada a sua data, o preço final por unidade do produto (visto que o preço pode ser negociado), a quantidade adquirida e o preço total pago, que é o resultado do produto dos dois valores anteriores. Um utilizador não pode efetuar compras das suas próprias ofertas. Sempre que é efetuada uma compra, a quantidade de produto em *stock* da respetiva oferta deve ser decrementada pela quantidade de produtos comprados. Após realizar uma compra, o comprador poderá, opcionalmente, fazer uma avaliação desta, atribuindo uma classificação quantitativa com base na sua satisfação, de uma a cinco estrelas, e acrescentando um comentário se assim o entender. Esta classificação irá refletir-se na classificação média do vendedor que publicou a oferta, que é a média de todas as classificações feitas a compras das suas ofertas.

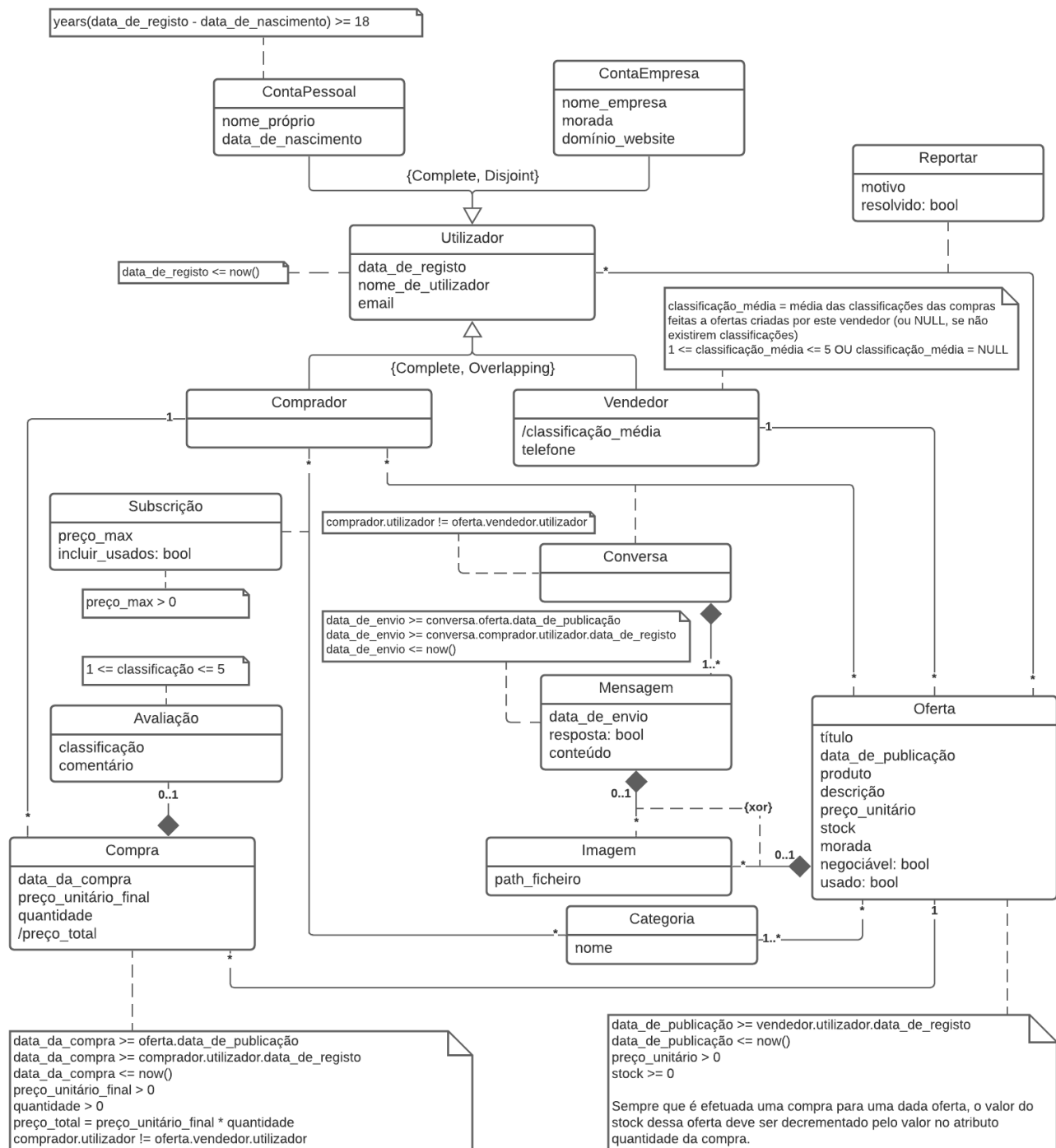
Um comprador poderá ainda assinalar interesse numa categoria de ofertas, efetuando uma subscrição, de modo a receber notificações via e-mail de novas ofertas que surjam pertencentes a essa categoria. Uma subscrição pode ser parametrizada por um preço máximo para o qual se está disposto a receber ofertas, e se há ou não interesse em produtos usados.

É bastante provável que o comprador necessite de entrar em contacto com o vendedor de uma oferta, seja para clarificar algum aspeto, negociar o preço ou combinar a entrega do produto (por exemplo, um local alternativo ao estipulado pela oferta, ou mesmo entrega por correio). Como tal, a plataforma implementa um sistema de mensagens. Um comprador pode iniciar uma conversa relativa a uma determinada oferta com o respetivo vendedor, enviando uma mensagem. Esta conversa incluirá todas as mensagens entre o comprador e o vendedor que sejam relativas a essa oferta em específico. Um utilizador não pode começar uma conversa relativa às suas próprias ofertas. De cada mensagem é necessário saber a data de envio, o conteúdo dela e se é enviada pelo comprador ou se é uma resposta do vendedor. Uma mensagem poderá incluir imagens anexadas, que são tratadas de modo idêntico ao já referido para as ofertas.

# Diagrama UML



## Diagrama UML Revisto



# Esquema Relacional

Utilizador(nome\_de\_utilizador, email, data\_de\_registo)  
{email} UK

ContaPessoal(utilizador->Utilizador, nome proprio, data\_de\_nascimento)

ContaEmpresa(utilizador->Utilizador, nome\_empresa, morada, dominio\_website)  
{dominio\_website} UK

Comprador(utilizador->Utilizador)

Vendedor(utilizador->Utilizador, classificacao\_media, telefone)

Oferta(id, vendedor->Vendedor, titulo, data\_de\_publicacao, produto, descricao, preco\_unitario, stock, morada, negociavel, usado)

Reportar(utilizador->Utilizador, oferta->Oferta, motivo, resolvido)

Conversa(comprador->Comprador, oferta->Oferta)

Mensagem(id, comprador->Conversa.comprador, oferta->Conversa.oferta, data\_de\_envio, resposta, conteudo)

Imagem(path\_ficheiro, mensagem->Mensagem, oferta->Oferta)

Categoria(nome)

CategoriaOferta(oferta->Oferta, categoria->Categoria)

Subscricao(comprador->Comprador, categoria->Categoria, preco\_max, incluir\_usados)

Compra(id, comprador->Comprador, oferta->Oferta, data\_da\_compra, preco\_unitario\_final, quantidade, preco\_total)

Avaliacao(compra->Compra, classificacao, comentario)

# Análise de dependências funcionais e formas normais

Para brevidade, apenas foram indicadas dependências funcionais completamente não triviais, visto que apenas essas são relevantes para verificar se há violações à 3ª Forma Normal ou à Forma Normal Boyce-Codd.

---

## Utilizador

$\{\text{nome\_de\_utilizador}\} \rightarrow \{\text{email}, \text{data\_de\_registo}\}$   
 $\{\text{email}\} \rightarrow \{\text{nome\_de\_utilizador}, \text{data\_de\_registo}\}$

Note-se que  $\{\text{nome\_de\_utilizador}\}$  e  $\{\text{email}\}$ , que aparecem do lado esquerdo das dependências funcionais indicadas, são chaves da relação.

## ContaPessoal

$\{\text{utilizador}\} \rightarrow \{\text{nome\_proprio}, \text{data\_de\_nascimento}\}$

Note-se que  $\{\text{utilizador}\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.

## ContaEmpresa

$\{\text{utilizador}\} \rightarrow \{\text{nome\_empresa}, \text{morada}, \text{dominio\_website}\}$   
 $\{\text{dominio\_website}\} \rightarrow \{\text{utilizador}, \text{nome\_empresa}, \text{morada}\}$

Note-se que  $\{\text{utilizador}\}$  e  $\{\text{dominio\_website}\}$ , que aparecem do lado esquerdo das dependências funcionais indicadas, são chaves da relação.

## Comprador

Não há dependências funcionais não triviais para esta relação.

## Vendedor

$\{\text{utilizador}\} \rightarrow \{\text{classificacao\_media}, \text{telefone}\}$

Note-se que  $\{\text{utilizador}\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.

## Oferta

$\{\text{id}\} \rightarrow \{\text{vendedor}, \text{titulo}, \text{data\_de\_publicacao}, \text{produto}, \text{descricao}, \text{preco\_unitario}, \text{stock}, \text{morada}, \text{negociavel}, \text{usado}\}$

Note-se que  $\{\text{id}\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.

## Reportar

$\{\text{utilizador}, \text{oferta}\} \rightarrow \{\text{motivo}, \text{resolvido}\}$

Note-se que  $\{\text{utilizador}, \text{oferta}\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.

## Conversa

Não há dependências funcionais não triviais para esta relação.

## Mensagem

$\{\text{id}\} \rightarrow \{\text{comprador}, \text{oferta}, \text{data\_de\_envio}, \text{resposta}, \text{conteudo}\}$

Note-se que  $\{\text{id}\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.



## Imagem

$\{path\_ficheiro\} \rightarrow \{mensagem, oferta\}$

Note-se que  $\{path\_ficheiro\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.

## Categoria

Não há dependências funcionais não triviais para esta relação.

## CategoriaOferta

Não há dependências funcionais não triviais para esta relação.

## Subscricao

$\{comprador, categoria\} \rightarrow \{preco\_max, incluir\_usados\}$

Note-se que  $\{comprador, categoria\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.

## Compra

$\{id\} \rightarrow \{comprador, oferta, data\_da\_compra, preco\_unitario\_final, quantidade, preco\_final\}$

$\{preco\_unitario\_final, quantidade\} \rightarrow \{preco\_total\}$

$\{preco\_unitario\_final, preco\_total\} \rightarrow \{quantidade\}$

$\{quantidade, preco\_total\} \rightarrow \{preco\_unitario\_final\}$

Do lado esquerdo das dependências funcionais indicadas aparece  $\{id\}$ ,  $\{preco\_unitario\_final, quantidade\}$ ,  $\{preco\_unitario\_final, preco\_total\}$  e  $\{quantidade, preco\_total\}$ . Destes, apenas  $\{id\}$  é uma chave. As três últimas dependências funcionais são, portanto, violações à Forma Normal Boyce-Codd, já que não são triviais e o lado esquerdo não é uma chave nem superchave. Como  $preco\_unitario\_final$ ,  $quantidade$  e  $preco\_total$  não são membros de uma chave candidata, não são atributos primos. Então, estas dependências funcionais também constituem violações à 3ª Forma Normal, uma vez que, para além de serem violações à Forma Normal de Boyce-Codd, o seu lado direito contém atributos não primos. Assim, a relação **Compra** não está nem na Forma Normal Boyce-Codd nem na 3ª Forma Normal.

No entanto, tendo em conta que estas violações ocorrem devido ao atributo derivado  $preco\_total$ , que é o produto de  $preco\_unitario\_final$  e  $quantidade$ , não se considera vantajosa a decomposição da relação para uma destas formas normais. De facto, tal decomposição implicaria uma relação com apenas estes três atributos, o que é contra-produtivo, visto que o propósito do atributo derivado é permitir obter o preço total da compra de forma mais prática. Estando noutra relação, essa vantagem é anulada e seria preferível simplesmente proceder à multiplicação direta dos outros dois atributos, evitando-se a relação extra. Para mais, se este atributo for implementado com uma *VIEW*, deixará de estar na tabela original, pelo que as dependências funcionais que constituem violações às duas formas normais também deixarão de existir: não faria sentido decompor a relação com base em dependências funcionais que vão deixar de existir. Assim, optou-se por não efetuar a decomposição da relação **Compra**. Note-se que, uma vez que a implementação da *VIEW* não era pretendida na 3ª entrega, não foi realizada.

## Avaliacao

$\{compra\} \rightarrow \{classificacao, comentario\}$

Note-se que  $\{compra\}$ , que aparece do lado esquerdo da dependência funcional indicada, é chave da relação.

---

Conclui-se assim que, com exceção do caso indicado explicitamente (relação **Compra**), não existem violações à Forma Normal Boyce-Codd, uma vez que, para todas as dependências funcionais não triviais nas restantes relações, o lado esquerdo é uma chave ou superchave. Como todas as relações que obedecem à Forma Normal Boyce-Codd também obedecem à 3ª Forma Normal (uma vez que a Forma Normal Boyce-Codd é mais estrita que a 3ª Forma Normal, ou seja, por definição, implica cumprir todas as condições da 3ª Forma Normal além das condições específicas da Forma Normal Boyce-Codd), conclui-se também que não há violações à 3ª Forma Normal, salvo no caso já referido.

# Restrições

Nota sobre algumas escolhas de tipos de dados recorrentes nas várias tabelas:

- Booleanos são guardados como inteiros, admitindo apenas valor 1 (Verdadeiro) ou 0 (Falso).
- Datas são guardadas como inteiros: o número de segundos que se passaram desde 1 de Janeiro de 1970 (*Unix Time*).
- Preços são guardados como inteiros, em cêntimos.

Relativamente às cláusulas *ON DELETE* das restrições de integridade referencial, decidiu-se por prudência restringir conservativamente a eliminação (em vez de eliminar em cascata), a menos nos casos em que o comportamento esperado fosse obviamente eliminar em cascata. Com isto pretende-se evitar perdas de dados acidentais com a eliminação de registos dos quais dependem muitos outros registos (que seriam eliminados em cascata, diretamente pela eliminação deste, ou pela eliminação de outros registos dos quais dependem e que dependem do primeiro).

Uma vez que na terceira entrega apenas se pretendia a implementação de três gatilhos, foram selecionadas, de entre as restrições que não foram implementadas na segunda entrega pela necessidade do uso de gatilhos, as três que se considerou mais pertinentes para implementar. As restantes, apesar de estarem presentes na lista que se segue, não foram implementadas (sendo identificadas na lista como não implementadas).

## Utilizador

- Todos os utilizadores têm nome de utilizador e não há nomes de utilizador repetidos. (*restrição chave: PRIMARY KEY*)
- Não há utilizadores com o mesmo email. (*restrição chave: UNIQUE*)
- Todos os utilizadores têm email e data de registo. (*NOT NULL*)
- A data de registo não pode ser posterior à data atual. (*restrição baseada em atributo: CHECK*)
- Todos os utilizadores têm uma conta pessoal ou de empresa (mas não ambos). (*não implementado*)
- Todos os utilizadores têm de estar inscritos como comprador e/ou vendedor na plataforma. (*não implementado*)

## ContaPessoal

- Não há múltiplas contas pessoais para o mesmo utilizador. (*restrição chave: PRIMARY KEY*)
- “utilizador” identifica o utilizador da conta. (*restrição de integridade referencial: FOREIGN KEY*) Se o utilizador for eliminado, a sua conta pessoal também é eliminada. (*ON DELETE CASCADE*) Alteração do nome de utilizador deve refletir-se na sua conta pessoal. (*ON UPDATE CASCADE*)
- É obrigatório o preenchimento dos campos “nome próprio” e “data de nascimento”. (*NOT NULL*)
- Apenas é permitido o registo a indivíduos com mais de 18 anos. (*implementado com um gatilho*)

## ContaEmpresa

- Não há múltiplas contas de empresa para o mesmo utilizador. (*restrição chave: PRIMARY KEY*)
- “utilizador” identifica o utilizador da conta. (*restrição de integridade referencial: FOREIGN KEY*) Se o utilizador for eliminado, a sua conta de empresa também é eliminada. (*ON DELETE CASCADE*) Alteração do nome de utilizador deve refletir-se na sua conta de empresa. (*ON UPDATE CASCADE*)
- É obrigatório o preenchimento do campo “nome empresa”. (*NOT NULL*)
- Duas empresas não podem ter o mesmo website. (*restrição chave: UNIQUE*)

## Comprador

- Um utilizador não pode estar inscrito mais do que uma vez como comprador. (*restrição chave: PRIMARY KEY*)
- “utilizador” identifica o utilizador que se inscreveu como comprador. (*restrição de integridade referencial: FOREIGN KEY*) Se o utilizador for eliminado, a sua inscrição como comprador também é eliminada. (*ON DELETE CASCADE*) Alteração do nome de utilizador deve refletir-se na sua inscrição como comprador. (*ON UPDATE CASCADE*)

## Vendedor

- Um utilizador não pode estar inscrito mais do que uma vez como vendedor. (*restrição chave: PRIMARY KEY*)
- “utilizador” identifica o utilizador que se inscreveu como vendedor. (*restrição de integridade referencial: FOREIGN KEY*) Se o utilizador for eliminado, a sua inscrição como vendedor também é eliminada. (*ON DELETE CASCADE*) Alteração do nome de utilizador deve refletir-se na sua inscrição como vendedor. (*ON UPDATE CASCADE*)
- A classificação média do vendedor, se existir, tem de estar compreendida entre 1 e 5. (*restrição baseada em atributo: CHECK*)
- É obrigatório por parte dos vendedores fornecer um contacto telefónico. (*NOT NULL*)
- A classificação média corresponde à média das classificações das compras feitas a ofertas publicadas pelo vendedor (ou *NULL*, se ainda não existirem classificações). (*implementado com um gatilho*)

## Oferta

- A cada oferta está associado um ID único. (*restrição chave: PRIMARY KEY*)
- É obrigatório especificar o vendedor, título da oferta, data de publicação, produto, descrição, *stock*, morada associada, se a oferta é negociável e se é um produto usado. (*NOT NULL*)
- “vendedor” identifica o vendedor que publicou esta oferta. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome de utilizador do vendedor deve refletir-se nas suas ofertas. (*ON UPDATE CASCADE*) No entanto, por precaução, não é permitida a eliminação do vendedor se houver ofertas publicadas por ele, de modo a evitar grandes perdas de dados por acidente (*comportamento por omissão, sem cláusula ON DELETE*). Poderão ser eliminadas todas as suas ofertas explicitamente antes de eliminar o vendedor, se realmente isso for pretendido.
- O preço unitário de venda do produto tem de ser um valor positivo. (*restrição baseada em atributo: CHECK*)
- O *stock* da oferta não pode ser um valor negativo. (*restrição baseada em atributo: CHECK*)
- As informações de negociável e usado apenas admitem valores de verdadeiro (1) ou falso (0). (*restrição baseada em atributo: CHECK*)
- Se a oferta não for negociável, é obrigatório indicar um preço unitário. De outro modo, o preço unitário é opcional. (*restrição baseada em tuplo: CHECK*)
- A data de publicação da oferta não pode ser posterior à data atual. (*restrição baseada em atributo: CHECK*)
- A data de publicação da oferta tem de ser posterior à data de registo do vendedor que a publicou. (*não implementado*)
- Sempre que é efetuada uma compra a uma dada oferta, o valor do *stock* dessa oferta deve ser decrementado pela quantidade comprada. (*implementado com um gatilho*)

## Reportar

- Cada utilizador só pode reportar uma determinada oferta uma vez. (*restrição chave: PRIMARY KEY*)
- “utilizador” identifica o utilizador que reportou. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome de um utilizador deve refletir-se nos *reports* que ele fez. (*ON UPDATE CASCADE*) No entanto, por precaução, não é permitida a eliminação do utilizador se houver *reports* feitos por ele, de modo a que estes não sejam eliminados quando talvez ainda se pretendesse analisá-los (*comportamento por omissão, sem cláusula ON DELETE*). Poderão ser eliminados todos os *reports* de um utilizador explicitamente antes de o eliminar, se se pretender eliminar o utilizador.
- “oferta” identifica a oferta que foi reportada. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do *id* de uma oferta deve refletir-se nos *reports* feitos a esta. (*ON UPDATE CASCADE*) Se uma oferta for eliminada, os *reports* feitos a esta também o serão. (*ON DELETE CASCADE*)
- A informação de se *report* está ou não resolvido apenas admite valores de verdadeiro (1) ou falso (0). (*restrição baseada em atributo: CHECK*)
- É obrigatório especificar o motivo pelo qual se está a reportar e se o *report* está ou não resolvido. (*NOT NULL*)

## Conversa

- Há no máximo uma conversa por cada comprador acerca de determinada oferta. (*restrição chave: PRIMARY KEY*)
- “comprador” identifica o comprador que iniciou a conversa. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome de utilizador de um comprador deve refletir-se nas conversas que iniciou. (*ON UPDATE CASCADE*) No entanto, por precaução, não é permitida a eliminação do comprador se houver conversas iniciadas por ele, de modo a que estas não sejam eliminadas acidentalmente quando talvez ainda se pretendesse analisá-las (por diversos motivos, entre eles rever a conversa à procura de comportamento impróprio por parte de um vendedor cuja oferta foi reportada)

(*comportamento por omissão, sem cláusula ON DELETE*). Poderão ser eliminadas explicitamente todas as conversas iniciadas por um comprador antes de o eliminar, se se pretender eliminar o utilizador.

- “oferta” identifica a oferta acerca da qual a conversa se baseia. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do *id* de uma oferta deve refletir-se nos *reports* feitos a esta. (*ON UPDATE CASCADE*) No entanto, por precaução, não é permitida a eliminação da oferta se houver conversas referentes a ela, de modo a que estas não sejam eliminadas acidentalmente quando o seu conteúdo talvez ainda fosse relevante para os compradores que as iniciaram (*comportamento por omissão, sem cláusula ON DELETE*). Poderão ser eliminadas todas as conversas relativas a uma oferta explicitamente antes de a eliminar, se se pretender realmente eliminar a oferta.
- Um utilizador não pode iniciar uma conversa relativa a uma oferta publicada pelo próprio. (*não implementado*)

## Mensagem

- A cada mensagem está associado um ID único. (*restrição chave: PRIMARY KEY*)
- “comprador” e “oferta” identificam a conversa a que a mensagem pertence. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome de utilizador do comprador ou do *id* da oferta relativos à conversa à qual a mensagem pertence devem refletir-se na mensagem. (*ON UPDATE CASCADE*) Se uma conversa for eliminada, todas as suas mensagens também o serão. (*ON DELETE CASCADE*)
- É obrigatório saber a que conversa uma mensagem pertence, a sua data de envio, o seu conteúdo e se se trata de uma resposta do vendedor. (*NOT NULL*)
- A informação de se a mensagem é uma resposta do vendedor (em vez de ter sido enviada pelo comprador) apenas admite valores de verdadeiro (1) ou falso (0). (*restrição baseada em atributo: CHECK*)
- A data de envio da mensagem não pode ser posterior à data atual. (*restrição baseada em atributo: CHECK*)
- A data de envio da mensagem tem de ser posterior à data de publicação da oferta e à data de registo do comprador interessado. (*não implementado*)

## Imagem

- Uma imagem pode ser unicamente identificada pelo *path* do respetivo ficheiro. (*restrição chave: PRIMARY KEY*)
- Uma imagem pertence sempre a uma mensagem ou a uma oferta (mas nunca a ambos). (*restrição baseada em tuplo: CHECK*)
- “mensagem” identifica a que mensagem esta imagem pertence, se pertencer a uma. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do *id* da mensagem à qual a imagem pertence deve refletir-se nesta. (*ON UPDATE CASCADE*) Se uma mensagem for eliminada, todas as suas imagens também o serão. (*ON DELETE CASCADE*)
- “oferta” identifica a que oferta esta imagem pertence, se pertencer a uma. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do *id* da oferta à qual a imagem pertence deve refletir-se nesta. (*ON UPDATE CASCADE*) Se uma oferta for eliminada, todas as suas imagens também o serão. (*ON DELETE CASCADE*)

## Categoria

- Todas as categorias têm um nome e duas categorias não podem ter o mesmo. (*restrição chave: PRIMARY KEY*)

## CategoriaOferta

- Uma oferta não pode estar associada à mesma categoria múltiplas vezes. (*restrição chave: PRIMARY KEY*)
- “oferta” determina a oferta desta associação. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do *id* da oferta deve refletir-se na associação. (*ON UPDATE CASCADE*) Se uma oferta for eliminada, todas as suas associações a categorias também o serão. (*ON DELETE CASCADE*)
- “categoria” determina categoria desta associação. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome da categoria deve refletir-se na associação. (*ON UPDATE CASCADE*) Se uma categoria for eliminada, todas as suas associações a ofertas também o serão. (*ON DELETE CASCADE*)
- Todas as ofertas têm de estar associadas a pelo menos uma categoria. (*não implementado*)

## Subscrição

- Um comprador não pode subscrever a mesma categoria múltiplas vezes. (*restrição chave: PRIMARY KEY*)
- “comprador” determina o comprador que efetuou a subscrição. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome de utilizador do comprador que efetuou a subscrição deve refletir-se na subscrição. (*ON UPDATE CASCADE*) Se um comprador for eliminado, todas as suas subscrições também o serão. (*ON DELETE CASCADE*)
- “categoria” determina a categoria subscrita. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome de uma categoria deve refletir-se nas subscrições feitas a esta. (*ON UPDATE CASCADE*) No entanto, por precaução, não é permitida a eliminação de uma categoria se houver subscrições referentes a ela, de modo a que estas não sejam eliminadas acidentalmente sem sequer notificar os compradores que subscreveram à categoria (*comportamento por omissão, sem cláusula ON DELETE*). Poderão ser eliminadas todas as subscrições de uma categoria explicitamente antes de a eliminar, se se pretender realmente eliminar a categoria.
- O preço máximo em que o comprador está interessado terá de ser positivo. (*restrição baseada em atributo: CHECK*)
- A informação de incluir ou não ofertas com produtos usados apenas admite valores de verdadeiro (1) ou falso (0). (*restrição baseada em atributo: CHECK*)

## Compra

- A cada compra está associado um ID único. (*restrição chave: PRIMARY KEY*)
- “comprador” identifica o comprador que efetuou a compra. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do nome de utilizador do comprador que efetuou uma compra deve refletir-se nesta. (*ON UPDATE CASCADE*) No entanto, por precaução, não é permitida a eliminação do comprador se este tiver efetuado compras, de modo a que estas não sejam eliminadas acidentalmente quando ainda são relevantes, por exemplo, para o vendedor da oferta (*comportamento por omissão, sem cláusula ON DELETE*). Poderão ser eliminadas explicitamente todas as compras de um comprador antes de o eliminar, se realmente se pretender eliminar o comprador.
- “oferta” identifica a oferta à qual foi efetuada a compra. (*restrição de integridade referencial: FOREIGN KEY*) Alteração do *id* de uma oferta deve refletir-se nas compras a ela efetuadas. (*ON UPDATE CASCADE*) No entanto, por precaução, não é permitida a eliminação de uma oferta se houver compras a ela efetuadas, de modo a que estas não sejam eliminadas acidentalmente quando ainda são relevantes, por exemplo, para o comprador (*comportamento por omissão, sem cláusula ON DELETE*). Poderão ser eliminadas explicitamente todas as compras a uma oferta antes de eliminar essa oferta, se realmente eliminar a oferta for o pretendido.
- É obrigatório estar indicado o comprador, a oferta, a data de compra, o preço unitário final, a quantidade e o preço total pago. (*NOT NULL*)
- O preço unitário final tem de ser positivo. (*restrição baseada em atributo: CHECK*)
- A quantidade comprada tem de ser positiva. (*restrição baseada em atributo: CHECK*)
- O preço total pago é o produto entre o preço unitário final e a quantidade comprada. (*restrição baseada em tuplo: CHECK*)
- A data de compra não pode ser posterior à data atual. (*restrição baseada em atributo: CHECK*)
- A data de compra tem de ser posterior à data de publicação da oferta e à data de registo do comprador. (*não implementado*)
- Um utilizador não pode comprar produtos das próprias ofertas. (*não implementado*)

## Avaliação

- Só é permitida no máximo uma avaliação por compra. (*restrição chave: PRIMARY KEY*)
- “compra” identifica a compra para a qual foi feita a avaliação. (*restrição de integridade referencial: FOREIGN KEYS*) Alteração do *id* da compra deve refletir-se na avaliação. (*ON UPDATE CASCADE*) Se uma compra for eliminada, todas as suas avaliações também o serão. (*ON DELETE CASCADE*)
- Uma avaliação possui necessariamente uma classificação. (*NOT NULL*)
- A classificação tem de estar compreendida entre 1 e 5. (*restrição baseada em atributo: CHECK*)

# Interrogações

Segue-se a lista ordenada das 10 interrogações implementadas, em linguagem natural:

**1.** Contar o número de compradores subscritos em cada categoria, ordenando os resultados pelo número de compradores em ordem decrescente. Incluir categorias sem subscrições.

Esta interrogação permite saber, para cada categoria, se esta é muito ou pouco procurada na plataforma.

**2.** Para cada vendedor que tenha recebido pelo menos um *report* numa das suas ofertas, listar o número total de *reports* feitos às suas ofertas e respetivo número de resolvidos e não resolvidos, ordenados desde os vendedores com mais *reports* por resolver até aos que têm menos, ou, em caso de empate, por ordem decrescente do total de *reports*.

É pertinente identificar os vendedores com mais *reports* por resolver, uma vez que é provável que estes tenham uma maior prioridade para ser resolvidos.

**3.** Listar os compradores que tenham feito compras de pelo menos um produto de cada categoria, isto é, que para todas as categorias haja uma oferta dessa categoria à qual tenham efetuado uma compra.

Esta interrogação permite identificar compradores com interesses mais diversificados, e que, por isso, poderão vir a interessar-se numa grande parte das ofertas disponíveis na plataforma.

**4.** Obter o dinheiro total ganho na plataforma (em euros) por cada vendedor, indicando também se a sua conta é uma conta pessoal ou uma conta de empresa.

Esta interrogação permite analisar e comparar o dinheiro total ganho na plataforma de vários vendedores, distinguindo as empresas de contas individuais, uma vez que é provável que isso tenha influência no modo como usam a plataforma e no número e características das ofertas que fazem, o que se reflete no dinheiro total ganho na plataforma.

**5.** Das mensagens de uma dada conversa (neste caso em concreto, entre a compradora *joana\_silva* e a conversa de *id 6*), listar por ordem cronológica: a data, quem enviou, quem recebeu, o conteúdo e o número de imagens anexadas.

Esta interrogação permite a leitura de uma conversa de forma mais clara.

**6.** Listar os utilizadores que tenham enviado mais de 5 mensagens e recebido mais de 5 mensagens no sistema de mensagens da plataforma.

Esta interrogação permite avaliar quais os utilizadores que fazem uso do sistema de mensagens da plataforma, de modo a determinar se este sistema é muito ou pouco usado e se faz sentido a sua manutenção.

**7.** Obter a lista de ofertas que tenham stock > 0 e que são compatíveis com as subscrições de um utilizador (neste caso em concreto, *pedro1987*), ordenadas pela classificação do respetivo vendedor (ou, em caso de empate, pela publicada mais recentemente). A listagem deve incluir informação do *id* da oferta, o seu título, produto, vendedor (com classificação média e telefone respetivos), descrição, preço unitário, stock, morada, se o preço é negociável, se se trata de um produto usado e a data de publicação.

Esta interrogação permite identificar ofertas que possam ser do interesse do comprador em questão, e que poderão eventualmente ser incluídas na notificação periódica via e-mail das suas subscrições.

**Nota:** para a correta indicação da classificação média de cada vendedor, deverá ser criado o primeiro gatilho definido na secção seguinte (correndo o ficheiro *gatilho1\_adiciona.sql*) antes de correr o ficheiro *povoar.sql*.

**8.** Obter a oferta com mais compras feitas no mês de setembro de 2018 e o seu vendedor, indicando também o respetivo número de compras nesse mês. Se mais do que uma oferta tiver o maior número de compras, todas deverão ser incluídas.

Esta interrogação permite saber qual a oferta mais “popular” num dado mês, em termos de compras.

**9.** Listar os vendedores que tenham respondido a todas as conversas das suas ofertas pelo menos uma vez. Vendedores sem conversas, uma vez que não deixaram conversas por responder, devem ser incluídos.

Esta interrogação ajuda a identificar vendedores que não deixaram questões de potenciais compradores por responder e, portanto, que possivelmente serão mais fáceis de contactar, o que leva a uma melhor experiência dos compradores.

**10.** Listar, para um dado comprador (neste caso em concreto, *miguel37*), o número de compras que fez em cada ano desde que se registou, bem como o gasto total em euros em cada um desses anos. Devem ser incluídos os anos em que o comprador não fez nenhuma compra.

Esta interrogação ajuda a perceber, para um dado comprador, a sua atividade na plataforma ao longo dos anos.

# Gatilhos

Segue-se a lista ordenada dos 3 gatilhos implementados, em linguagem natural:

1. Quando é feita uma avaliação na sequência da realização de uma compra, o vendedor que publicou a oferta à qual foi feita a compra deverá ter a sua classificação média atualizada, de modo a refletir a classificação da nova avaliação feita.

Foi escolhido implementar o gatilho para a inserção de novas avaliações na base de dados. Porém, a classificação média do vendedor deveria ser atualizada também se uma das avaliações feitas às suas ofertas for atualizada (*AFTER UPDATE ON Avaliacao*) ou eliminada (*AFTER DELETE ON Avaliacao*). Para isso, podiam ser implementados gatilhos semelhantes, mas dos tipos indicados (*AFTER UPDATE* ou *AFTER DELETE*).

2. O *stock* de uma oferta deve ser atualizado sempre que é feita uma compra a essa oferta, tendo em conta a quantidade de produtos comprada.

Foi escolhido implementar o gatilho para a inserção de novas compras na base de dados. Também poderia ser implementado um gatilho para tratar da atualização da quantidade de compras existentes (*AFTER UPDATE ON Compra*), uma vez que esta atualização seria feita apenas para correção da compra se tivesse sido inserida uma quantidade errada. Porém, não se considera que se devesse implementar um gatilho para tratar da situação em que fosse eliminada uma compra, uma vez que não é óbvio se a compra foi eliminada por que se tratava de um erro, ou se simplesmente se pretendia apagar o seu registo (e neste último caso, não faz sentido alterar o *stock*). Se fosse pretendido, o *stock* poderia ser atualizado manualmente após apagar a compra.

3. Deve-se impedir a criação de contas pessoais para quem não tenha a idade mínima de 18 anos.

Foi escolhido implementar o gatilho para a inserção de novas contas pessoais. Porém, a restrição que este gatilho implementa também pode ser violada no caso da atualização da data de nascimento de uma conta pessoal ou no caso da atualização da data de registo de um utilizador (embora, em princípio, não haja motivo para fazer a atualização da data de registo, a menos que seja para correção de um erro). Para garantir que a restrição não é violada nesses casos, deveriam ser implementados gatilhos *BEFORE UPDATE ON ContaPessoal* e *BEFORE UPDATE ON Utilizador*, respetivamente.

**Nota:** Esta restrição foi testada (no ficheiro *gatilho3\_verifica.sql*) com três casos. Nos dois primeiros, são criados utilizadores maiores de 18 anos, pelo que a restrição não é violada. Já no último, tenta-se criar uma conta pessoal que violaria a restrição, pelo que o gatilho desencadeia um erro que impede o registo na tabela *ContaPessoal*. No entanto, a entrada na tabela *Utilizador*, que já havia sido adicionada, não é removida. Poderia ser feita a criação do utilizador e da conta dentro de uma mesma transação para garantir que, havendo um erro, o novo utilizador não fosse registado apenas parcialmente. No entanto, o uso de transações não era pretendido para a terceira entrega do projeto.

Deverão ser adicionados os gatilhos antes de correr o ficheiro *povoar.sql*, caso se pretenda correr esse ficheiro, de modo a garantir a coerência e a atualização correta dos dados (em especial, da classificação média do vendedor, que implica a presença do primeiro gatilho).