Base de Dados

Loja de Hardware

Licenciatura em Engenharia Informática e Computação

*2021/2022*

**Turma 2 - Grupo 202**

Alberto Serra - up202103627

Eduardo Correia - up201909914

José Carvalho - up202005827

**Professor**

Lázaro Costa

**Índice**

[1.Contexto 2](#_Toc92920691)

[2.Diagrama UML 3](#_Toc92920692)

[3.Esquema Relacional 4](#_Toc92920693)

[4.Análise de Dependências Funcionais e Formas Normais 5](#_Toc92920694)

[4.1 Relação Cliente 5](#_Toc92920695)

[4.2 Relação Venda 5](#_Toc92920696)

[4.3 Relação Reparação 5](#_Toc92920697)

[4.4 Relação Funcionário 5](#_Toc92920698)

[4.5 Relação Salário Mensal 6](#_Toc92920699)

[4.6 Relação Especialidade 6](#_Toc92920700)

[4.7 Relação Incremento Venda 6](#_Toc92920701)

[4.8 Relação Incremento Reparação 6](#_Toc92920702)

[4.9 Relação Peça 6](#_Toc92920703)

[4.10 Relação Peça Reparação 7](#_Toc92920704)

[4.11 Relação Peça Fornecedor 7](#_Toc92920705)

[4.12 Relação Peça Venda 7](#_Toc92920706)

[4.13 Relação Fornecedor 7](#_Toc92920707)

[4.14 Relação Armazém 7](#_Toc92920708)

[5.Implementação de restrições na Base de dados 8](#_Toc92920709)

[5.1 Tabela Cliente 8](#_Toc92920710)

[5.2 Tabela Venda 8](#_Toc92920711)

[5.3 Tabela Reparação 8](#_Toc92920712)

[5.5 Tabela Salário Mensal 9](#_Toc92920713)

[5.8 Tabela Incremento Reparação 10](#_Toc92920714)

[5.9 Tabela Peça 10](#_Toc92920715)

[5.10 Tabela Peça Reparação 10](#_Toc92920716)

[5.11 Tabela Peça Fornecedor 10](#_Toc92920717)

[5.12 Tabela Peça Venda 11](#_Toc92920718)

[5.13 Tabela Fornecedor 11](#_Toc92920719)

[5.14 Tabela Armazém 11](#_Toc92920720)

[6.Divisão do projeto pelos elementos do grupo 12](#_Toc92920721)

# **1.Contexto**

No âmbito de satisfazer os objetivos propostos no projeto, é pretendido a criação de uma base de dados para uma loja de hardware de modo que esta possa informatizar os seus registos. Para tal, é necessário o registo de dados relativos aos seus clientes, o serviço que estes pretendem realizar, a gestão dos custos internos de mão de obra e da aquisição dos produtos, desde os seus fornecedores, à venda dos mesmos para os seus compradores.

De modo a contextualizar o modelo conceptual, é Importante em relação a cada cliente guardar o seu nome, NIF, morada, email e telemóvel. Estes podem usufruir de dois tipos de serviços, a venda e reparação de hardware.

Por cada serviço é armazenado o dia em que foi realizado, o seu custo e a designação. No caso de ser uma venda, é guardado o lucro produzido, o número de peças vendidas, a morada de carga e descarga do produto, devido ao facto que uma venda pode ou não ser feita presencialmente na loja. Se o serviço for uma reparação é guardado a data de início e final desta mesma, visto que um processo de conserto se pode estender por mais do que um dia.

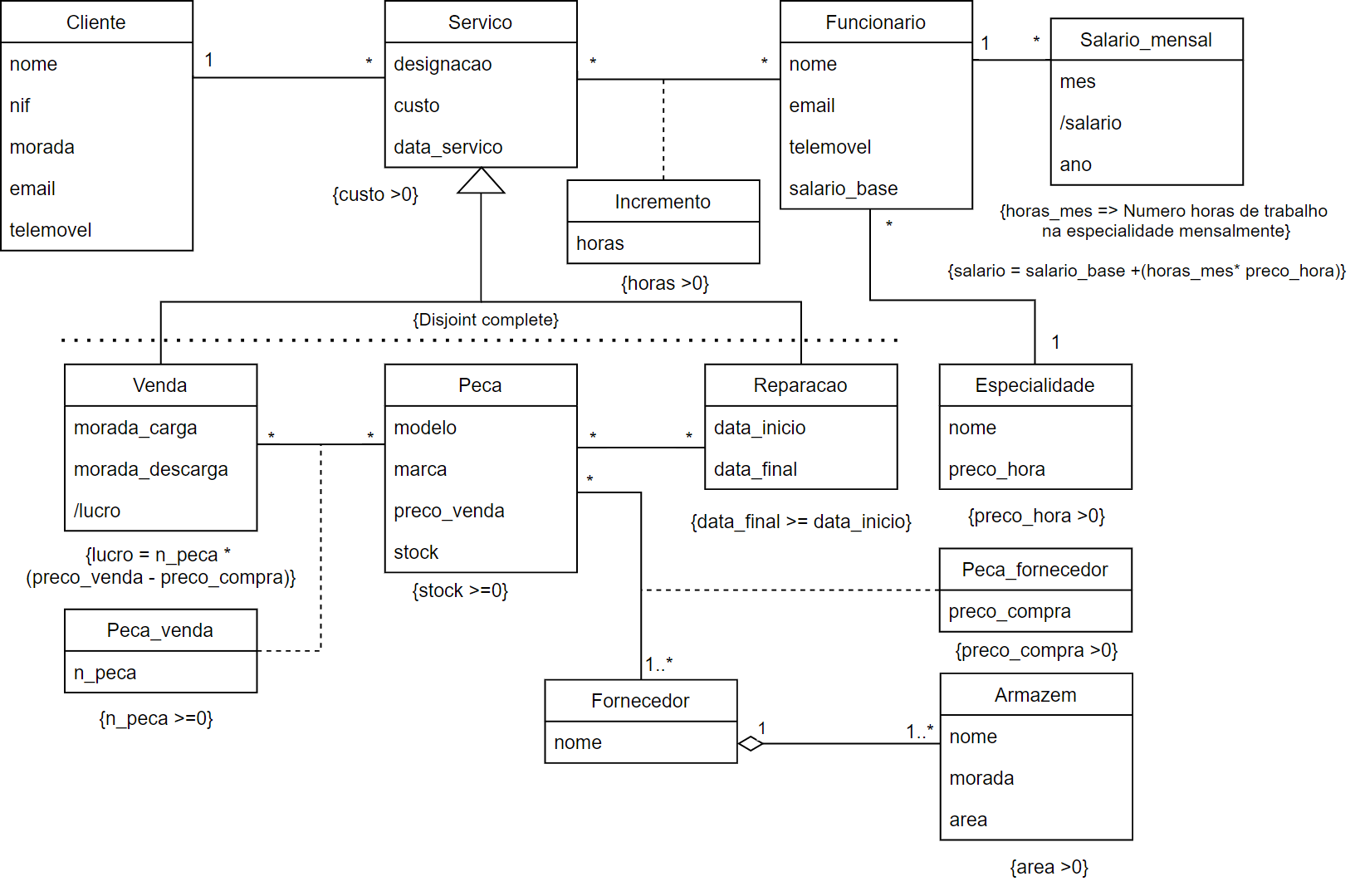
A loja guarda um registo de todas as peças de hardware disponíveis. Estas são utilizadas em ambos os serviços disponibilizados. Sobre cada peça sabe-se o modelo, a marca, a quantidade e o preço a que é vendida ao cliente.

Destas peças também sabemos a sua origem. Dos fornecedores em questão, sendo que uma peça pode ser comprada a diversos fornecedores e estes também possuírem as mesmas peças para venda, conhecemos os seus nomes e preço de venda, isto é, o custo de cada peça adquirida pela loja. Cada um destes tem acesso a diversos armazéns com nome, morada e área disponível para armazenamento.

Por fim, existem os funcionários do estabelecimento. Destes interessa saber o seu nome, e-mail, telemóvel, salário e horas prestadas em serviços. Sendo assim, cada um destes tem uma especialidade que determina o valor monetário que estes recebem por hora de trabalho, aliando à possibilidade de receberem acima de um valor base. Para tal efeito, é registado o salário mensal alusivo a cada um destes, associado ao respetivo mês e ano.

# **2.Diagrama UML**

Com a finalidade de estruturar e modelar visualmente a respetiva base de dados da loja de hardware, criou-se o seguinte diagrama UML (*Unified Modeling Language*), estabelecendo uma representação mais apelativa e *"user-friendly"* das inúmeras associações existentes entre classes e atributos. Este diagrama auxilia o raciocínio lógico necessário para a realização das próximas tarefas do projeto, facilitando a comunicação entre os múltiplos objetos (melhorando a compreensão esquemática do mesmo) e a identificação dos vários processos definidos.



# **3.Esquema Relacional**

Uma base de dados relacional permite, para além da organização objetiva e simples da informação, a criação de relações entre os grupos de informação distintos. A definição destas relações é feita com base na teoria dos conjuntos e lógica de predicados.

O Esquema Relacional, é adequado a ser o modelo subjacente de um Sistema de Gestão de Base de Dados, modelando um conjunto de elementos representativos num grupo de relações. Estas são constituídas por um conjunto de atributos que definem as propriedades relevantes da entidade que representam. Baseia-se então no princípio de que todos os dados estão armazenados em tabelas, onde cada *tuplo* (linha ou registo) da relação caracteriza um único objeto.

**Cliente** (idCliente, *nif*, nome, morada, email, telemovel)

**Venda** (*idVenda*, designacao, custo, data\_servico, morada\_carga, morada\_descarga, lucro, idCliente-> Cliente)

**Reparacao** (*idReparacao*, designacao, custo, data\_servico, data\_inicio, data\_final, idCliente -> Cliente)

**Funcionario** (*idFuncionario*, nome, morada, email, telemovel, salario\_base, idEspecialidade -> Especialidade)

**Salario\_mensal** (*idSalario\_mensal*, mes, salario, ano, idFuncionario -> Funcionario)

**Especialidade** (*idEspecialidade*, nome, preco\_Hora)

**IncrementoVenda** (*idVenda* -> Venda, *idFuncionario* -> Funcionario)

**IncrementoReparacao** (*idReparacao* -> Reparação, *idFuncionario* -> Funcionario, horas)

**Peca** (*idPeca*, modelo, marca, preco\_venda, stock)

**PecaReparacao** (*idPeca* -> Peca, *idReparacao* -> Reparacao)

**PecaFornecedor** (*idPeca* -> Peca, *idFornecedor* -> Fornecedor, preco\_compra)

**PecaVenda**(*idVenda* -> Venda, *idPeca* ->Peca, n\_peca)

**Fornecedor** (*idFornecedor*, nome)

**Armazem** (*idArmazem*, nome, morada, área, idFornecedor -> Fornecedor)

# 

# **4.Análise de Dependências Funcionais e Formas Normais**

## **4.1 Relação Cliente**

*Dependências funcionais:*

*idCliente -> nif, nome, morada, email, telemóvel*

*email -> idCliente, nif, nome, morada, telemóvel*

As chaves são o idCliente e o email. Embora o nif e o telemóvel sejam únicos para cada cliente, a inserção dos mesmos na base de dados será opcional, tendo em conta a informação que achamos relevante guardar, obrigatoriamente, de um cliente.

A relação Cliente encontra-se na forma normal de Boyce-Codd porque cada dependência funcional tem uma chave à sua esquerda, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial).

## **4.2 Relação Venda**

*Dependências funcionais:*

*idVenda -> designacao, custo, data\_servico, morada\_carga, morada\_descarga, lucro, idCliente*

Visto que a chave é o idVenda, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial), então a relação Venda encontra-se na forma normal de Boyce-Codd.

## **4.3 Relação Reparação**

*Dependências funcionais:*

*idReparacao -> designacao, custo, data\_servico, data\_inicio, data\_final, idCliente*

Visto que a chave é o idReparacao, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial), então a relação Reparação encontra-se na forma normal de Boyce-Codd.

## **4.4 Relação Funcionário**

Dependências funcionais:

*idFuncionario -> nome, morada, email, telemovel, salario\_base, idEspecialidade*

*email -> idFuncionario, nome, morada, telemovel, salario\_base, idEspecialidade*

As chaves são o idFuncionario e o email. Embora o telemóvel seja único para cada cliente, a inserção do mesmo na base de dados será opcional, tendo em conta a informação que achamos relevante guardar, obrigatoriamente, de um funcionário.

A relação Funcionário encontra-se na forma normal de Boyce-Codd porque cada dependência funcional tem uma chave à sua esquerda, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial).

## **4.5 Relação Salário Mensal**

*Dependências funcionais:*

*idSalario\_mensal -> mes, salario, ano, idFuncionario*

Visto que a chave é o idSalario\_mensal, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial), então a relação Salário Mensal encontra-se na forma normal de Boyce-Codd.

## **4.6 Relação Especialidade**

*Dependências funcionais:*

*idEspecialidade -> nome, preco\_hora*

Visto que a chave é o idEspecialidade, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial), então a relação Salário Mensal encontra-se na forma normal de Boyce-Codd.

## **4.7 Relação Incremento Venda**

*Dependências funcionais:*

*idVenda, idFuncionario -> idVenda, idFuncionario*

Visto que temos uma chave composta por idVenda e idFuncionario, que se determina trivialmente, então a relação Incremento Venda encontra-se na forma normal de Boyce-Codd.

## **4.8 Relação Incremento Reparação**

*Dependências funcionais:*

*idReparacao, idFuncionario -> horas*

Visto que temos uma chave composta por idReparacao e idFuncionario, que determina o atributo horas (elementos na direita da dependência funcional mais os triviais), então a relação Incremento Venda encontra-se na forma normal de Boyce-Codd.

## **4.9 Relação Peça**

*Dependências funcionais:*

*idPeca -> modelo, marca, preco\_venda, stock*

Visto que a chave é o idPeca, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial), então a relação Peça encontra-se na forma normal de Boyce-Codd.

## **4.10 Relação Peça Reparação**

*Dependências funcionais:*

*idPeca, idReparacao -> idPeca, idReparacao*

Visto que temos uma chave composta por idPeca e idReparacao, que se determina trivialmente, então a relação Peça Reparação encontra-se na forma normal de Boyce-Codd.

## **4.11 Relação Peça Fornecedor**

*Dependências funcionais:*

*idPeca, idFornecedor -> preco\_compra*

Visto que temos uma chave composta por idPeca e idFornecedor, que determina o atributo preco\_compra (elemento na direita da dependência funcional mais o trivial), então a relação Peça Fornecedor encontra-se na forma normal de Boyce-Codd.

## **4.12 Relação Peça Venda**

*Dependências funcionais:*

*idVenda, idPeca -> n\_peca*

Visto que temos uma chave composta por idVenda e idPeca, que determina o atributo n\_peca (elemento na direita da dependência funcional mais o trivial), então a relação Peça Venda encontra-se na forma normal de Boyce-Codd.

## **4.13 Relação Fornecedor**

*Dependências funcionais:*

*idFornecedor -> nome, preco\_compra*

Visto que a chave é o idFornecedor, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial), então a relação Fornecedor encontra-se na forma normal de Boyce-Codd.

## **4.14 Relação Armazém**

*Dependências funcionais:*

*idArmazem -> nome, morada, área, idFornecedor*

Visto que a chave é o idArmazem, determinando funcionalmente todos os restantes elementos da relação (elementos na direita da dependência funcional mais o trivial), então a relação Armazém encontra-se na forma normal de Boyce-Codd.

# **5.Implementação de restrições na Base de dados**

De forma que a nossa base de dados esteja mais otimizada e menos suscetível ao erro na introdução de novos valores, é criado um conjunto de restrições que irão prevenir a inserção indesejada de informação nas relações esquematizadas em SQLite. Assim, torna-se possível manter a integridade dos dados armazenados e o padrão de elementos idealizados nas nossas tabelas.

## **5.1 Tabela Cliente**

A chave primária é o ID\_Cliente. Todos os atributos menos o Telemóvel são obrigatórios. O NIF, email e telemóvel são únicos.

* ID\_Cliente Primary Key,
* NIF Unique,
* Nome Not Null,
* Morada Not Null
* Email Unique Not Null,
* Telemovel Unique.

## **5.2 Tabela Venda**

A chave primária é o ID\_Venda. O ID\_cliente é uma chave estrangeira que referencia a tabela cliente. Todos os atributos com exceção da designação e do lucro são obrigatórios. O custo e o lucro são maiores que 0.

* ID\_Venda Primary Key,
* Custo Not Null,
* Data\_Serviço Not Null,
* Morada\_Carga Not Null,
* Morada\_Descarga Not Null,
* ID\_Cliente Not Null Foreign Key -> Cliente.

## **5.3 Tabela Reparação**

A chave primária é o ID\_Reparação. O ID\_cliente é uma chave estrangeira que referencia a tabela cliente. Todos os atributos com exceção da designação e do lucro são obrigatórios. A data final é superior à data de início, e o custo é maior que 0.

* ID\_Reparação Primary Key,
* Data\_Serviço Not Null,
* Data\_Inicio Not Null,
* Data\_Final Not null,
* ID\_Cliente Not Null Foreign Key -> Cliente.

**5.4 Tabela Funcionário**

A chave primária é o ID\_Funcionário. O ID\_Especialidade é uma chave estrangeira que referencia a tabela especialidade. Todos os atributos com exceção do telemóvel são obrigatórios.

* ID\_Funcionario Primary Key,
* Nome Not Null,
* Morada Not Null,
* Email Unique Not Null,
* Telemovel Unique Not Null,
* ID\_Especialidade Not Null Foreign Key -> Especialidade.

## **5.5 Tabela Salário Mensal**

A chave primária é ID\_Salario\_Mensal. O ID\_Funcionário é uma chave estrangeira que referencia a tabela funcionário. Todos os atributos são obrigatórios. É verificado que o mês está entre 1 e 12 e o salário é maior que 0.

* ID\_Salario\_Mensal Primary Key,
* Mes Not Null (1<=Mes<=12),
* Salario Not Null (>0),
* Ano Not Null,
* ID\_Funcionario Not Null Foreign Key -> Funcionario.

**5.6 Tabela Especialidade**

A chave primária é ID\_Especialidade. Todos os atributos são obrigatórios. É verificado se Preco\_Hora é maior que zero.

* ID\_Especialidade Primary Key,
* Nome Not Null,
* Preco\_Hora Not Null (>0)

**5.7 Tabela Incremento Venda**

A chave primária é a chave composta (ID\_Venda,ID\_Funcionario). O ID\_Venda é uma chave estrangeira que referencia a tabela venda e o ID\_Funcionario é uma chave estrangeira que referencia a tabela Funcionario. Todos os atributos são obrigatórios.

* ID\_Venda Primary Key Foreign Key -> Venda,
* ID\_Funcionario Primary Key Foreign Key -> Funcionario.

## **5.8 Tabela Incremento Reparação**

A chave primária é a chave composta (ID\_Reparacao,ID\_Funcionário). O ID\_Reparacao é uma chave estrangeira que referencia a tabela venda e o ID\_Funcionário é uma chave estrangeira que referencia a tabela Funcionário. Ambos os ID’s são obrigatórios. O atributo horas é por norma 0.

* ID\_Reparacao Primary Key Foreign Key -> Reparacao,
* ID\_Funcionario Primary Key Foreign Key -> Funcinario,
* Horas (Default = 0 e >0).

## **5.9 Tabela Peça**

A chave primária é o ID\_Peca. Por cada peca registada é guardado o modelo, o preco de venda assim como o stck da mesma, sendo todos os campos not null. Cada entrada tem um ID correspondente.

* Id\_Peca Primary Key,
* Modelo Not Null,
* Preco\_Venda Not Null (>0),
* Stock Not Null (>=0).

## **5.10 Tabela Peça Reparação**

A chave primária é a chave composta (ID\_Peca,ID\_Reparacao). O ID\_Peca é uma chave estrangeira que referencia a tabela peca e o ID\_Reparacao é uma chave estrangeira que referencia a tabela Reparacao. Ambos os ID’s são obrigatórios.

* ID\_Peca Primary Key Foreign Key -> Peca,
* ID\_Reparacao Primary Key Foreign Key -> Reparacao.

## **5.11 Tabela Peça Fornecedor**

A chave primária é a chave composta (ID\_Peca,ID\_Fornecedor). O ID\_Peca é uma chave estrangeira que referencia a tabela peca e o ID\_Fornecedor é uma chave estrangeira que referencia a tabela Fornecedor. Todos os atributos são obrigatórios. É verificado se o preço de compra é maior ou igual a zero.

* ID\_Peca Primary Key Foreign Key -> Peca,
* ID\_Fornecedor Primary Key Foreign Key -> Fornecedor,
* Preco\_Compra Not Null.

## **5.12 Tabela Peça Venda**

A chave primária é a chave composta (ID\_Peca,ID\_Venda). O ID\_Peca é uma chave estrangeira que referencia a tabela peca e o ID\_Venda é uma chave estrangeira que referencia a tabela venda. Todos os atributos são obrigatórios. É verificado se N\_Peca é maior ou igual a zero.

* ID\_Peca Primary Key Foreign Key -> Peca,
* ID\_Venda Primary Key Foreign Key -> Venda,
* N\_Peca Not Null.

## **5.13 Tabela Fornecedor**

A chave primária é o ID\_Fornecedor. Todos os atributos são obrigatórios.

* ID\_Fornecedor Primary Key,
* Nome Not Null.

## **5.14 Tabela Armazém**

A chave primária é ID\_Armazem. Todos os atributos são obrigatórios. O ID\_Fornecedor é uma chave estrangeira que referencia a tabela Fornecedor. É verificado se Area é maior que zero.

* ID\_Armazem Primary Key,
* Nome Not Null,
* Morada Not Null,
* Area Not Null,
* ID\_Fornecedor Not Null Foreign Key -> Fornecedor.

# **6.Divisão do projeto pelos elementos do grupo**

Todo o Trabalho foi dividido igualmente pelos três elementos. Cada elemento do grupo foi responsável por 33% de cada tarefa proposta, seja esta o context, o UML, definição do esquema relacional, restrições, dependências funcionais e formas normais, criação da base de dados e povoamento da mesma.

Assim sendo, cada elemento contribuiu para qualquer parte do projeto com uma divisão equitativa do mesmo.

**7.Observações sobre a segunda entrega**

Relativamente à segunda entrega, foram resolvidos todos os possíveis erros relativamente ao nosso esquema SQL e UML, adicionando as opções on delete e on update nas respetivas chaves estrangeiras. Além disso, a base de dados foi povoada com um maior número de dados, tornando esta mais robusta e realista. Assim será possivel utilizar as interrogações e gatilhos de forma mais pertinente.

**8.Lista das Interrogações**

Uma interrogação é um pedido de uma informação ou de um dado a uma respetiva base de dados. Este pedido ou consulta permite selecionar de forma exata e clara a informação que queremos visualizar de momento.

Neste ponto segue-se uma lista das interrogações propostas sobre a base de dados que foi desenvolvida. Foi tido em conta a relevância das respetivas interrogações face ao contexto e tema do trabalho. Algumas interrogações contêm a utilização de vistas para as tornar mais legíveis e menos repetitivas.

**1.** Listagem dos clientes com o maior número de compras efetuadas, juntamente com o total de valor gasto e o desconto acumulado pelos mesmos para ser utilizado numa próxima compra (7,5% do valor).

**2.** Listagem dos Funcionários que realizaram algum tipo de reparação e possuam um salário mensal igual ou superior a 1000 euros no ano de 2021.

**3.** Listagem dos Armazéns com o maior valor de área de um respetivo fornecedor (caso este tenha mais do que um armazém), com o número de peças disponíveis que têm o maior valor final e stock igual ou superior a 5 unidades.

**4.** Média dos salários dos Funcionários especialistas em computadores, excluindo os profissionais relacionados com a administração da loja.

**5.** Consulta das vendas online mais lucrativas aliadas ao nome do funcionário responsável pelas mesmas.

**6.** Quais as designações das reparações que demoram pelo menos dois dias a serem realizadas e o seu respetivo custo, juntamente do seu cliente e peça utilizada no mesmo serviço (caso tenha sido necessário).

**7.** Exibição dos dois funcionários mais produtivos da loja. O funcionário responsável pelo maior número de vendas unido ao funcionário com o maior número de reparações realizadas.

**8.** Lucro da loja desde a sua abertura até ao momento atual, contabilizando todo o tipo de despesas (sendo que o lucro pode ou não ser negativo).

**9.** Os fornecedores (caso existam mais do que um) responsáveis pelas peças mais utilizadas no serviço de reparação.

**10.** Enumeração do vencimento de cada funcionário por hora no mês de dezembro de 2021, aliado a um respetivo bónus relativo ao subsídio de Natal.

**9. Lista dos Gatilhos**

No processo de implementação dos gatilhos foi notado que para certas funcionalidades seriam necessários criar mais do que um para que os processos se otimizados e automáticos. Tal efeito seria bastante difícil de realizar com apenas um gatilho por ficheiro, e desta forma conseguimos produzir o resultado esperado e de uma forma mais legível e eficiente.

**1.** Quando a loja se encontra prestes a realizar uma venda de uma peça a um cliente, verifica se esta se encontra com uma quantidade de stock superior ou igual à quantidade pretendida pelo mesmo. Simultaneamente, dá-se a atualização do stock da peça caso esta tenha sido vendida com sucesso e encomenda-se uma quantidade de cinco unidades para que o stock nunca fique vazio. Assim, foi possível criar um sistema automático de gestão de stocks associado às vendas.

**2.** Ao adicionar um registo de um salário mensal, verifica se o respetivo funcionário já possui um valor de salário nesse respetivo mês. Simultaneamente, caso a inserção de salário seja possível, verifica se o valor deste é valido no contexto da base de dados.

**3.**