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](#_Toc90034641)

[2.Diagrama UML 3](#_Toc90034642)

[3.Esquema Relacional 4](#_Toc90034643)

[5.Implementação de restrições na Base de dados 5](#_Toc90034644)

[5.1 Identificação com incremento 5](#_Toc90034645)

[5.2 Tabela Clientes 5](#_Toc90034646)

[5.3 Tabela Venda 5](#_Toc90034647)

[5.4 Tabela Reparação 5](#_Toc90034648)

[5.5 Tabela Funcionário 6](#_Toc90034649)

[5.6 Tabela Salário Mensal 6](#_Toc90034650)

[5.7 Tabela Incremento Reparação 6](#_Toc90034651)

[5.8 Tabela Peça 6](#_Toc90034652)

[4.9 Tabela Peça Reparação 6](#_Toc90034653)

[5.10 Tabela Peça Fornecedor 7](#_Toc90034654)

[5.11 Tabela Peça Venda 7](#_Toc90034655)

[5.12 Tabela Fornecedor 7](#_Toc90034656)

[5.13 Tabela Armazém 7](#_Toc90034657)

# 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.

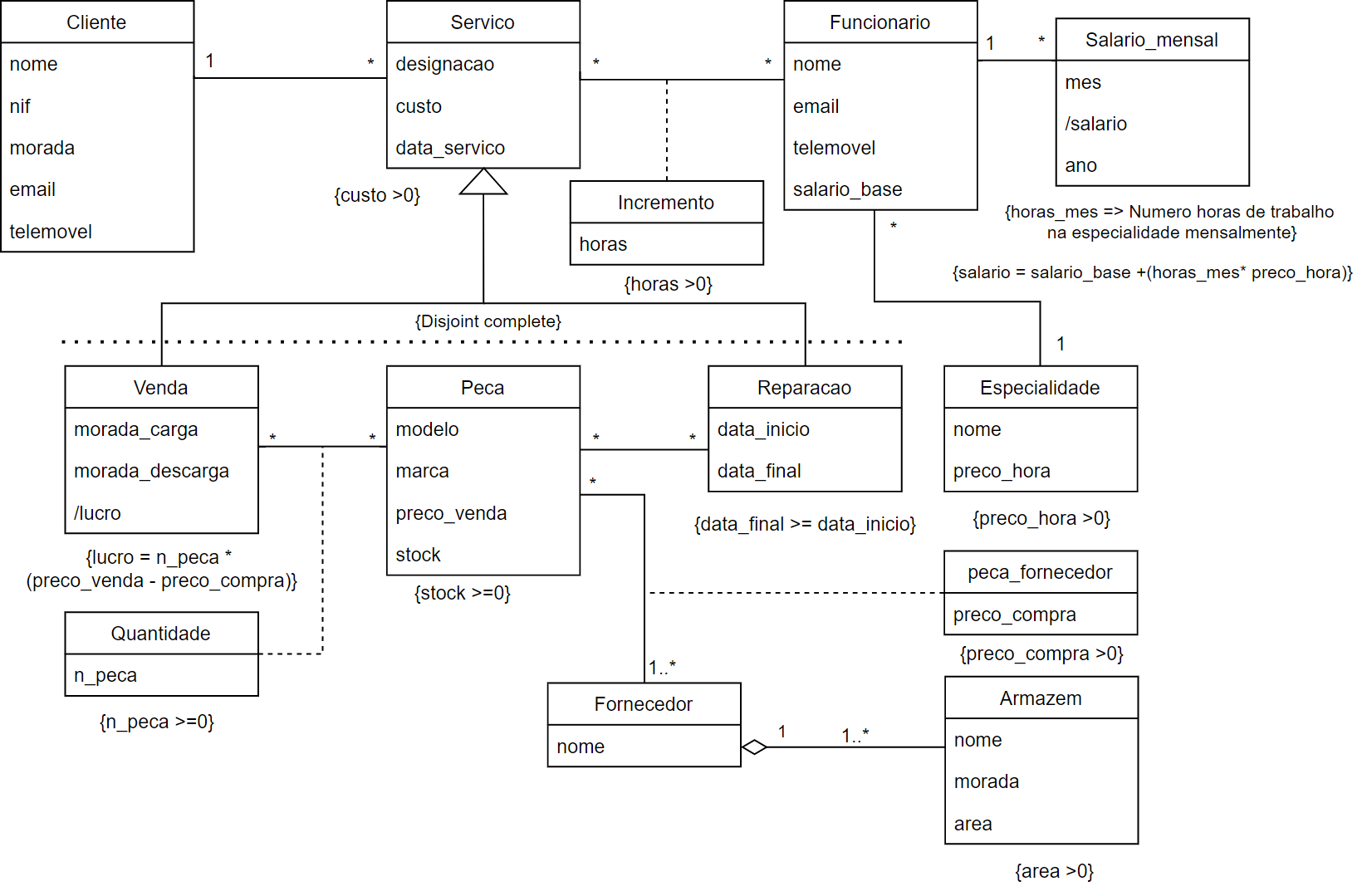
Destas peças também sabemos a sua origem. Dos fornecedores em questão, sendo que uma peça pode ser comprada a diversos fornecedores e estes também possuírem as mesmas peças para venda, conhecemos os seus nomes e preço de venda, isto é, o custo de cada peça adquirida pela loja. Cada um destes tem acesso 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* -> Venda, *idFuncionario* -> Funcionario, horas)

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

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

**PecaFornecedor** (*idPeca* -> Peca, *idFornecedor* -> Fornecedor)

**PecaVenda**(*idVenda* -> Venda, *idPeca* ->Peca, n\_peca) #perguntarAoStor<3(vs quantidade)

**Fornecedor** (*idFornecedor*, nome, preco\_compra)

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

# 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 Identificação com incremento

Por norma, todas as tabelas, com exceção das tabelas de relação, utilizam uma chave primária com o nome de *id\_<nomeTabela>* e com o campo de *auto incremento* para facilitar a inserção e garantir que não existe a repetição deste. Devido ao método de funcionamento do acréscimo automático e a forma como os dados são inseridos, não utilizamos o campo não nulo, para estas chaves, pois nunca aconteceria.

## 5.2 Tabela Clientes

Em relação aos clientes é necessário guardar o seu *identificador*, na forma de uma chave primária de *auto incremento*, o seu *NIF*, o seu *nome*, a *morada*, o *email* e opcionalmente o seu *telemóvel*. Com exceção deste último todos os atributos são obrigatórios. O *NIF*, *email* e *telemóvel* *tratam-se* de serem também todos *únicos*.

## 5.3 Tabela Venda

Nesta tabela guarda-se um *identificador*, na forma de uma chave primária de *auto incremento*, a designação de *venda*, o seu *custo*, a *data de serviço*, a *morada de carga*, a *morada de descarga*, o *lucro* e o *identificador* do cliente. O *custo*, *data de serviço*, *morada de carga*, *morada de descarga*, *lucro* e o *identificador* do cliente são todos atributos obrigatórios. O *custo* e *lucro* são ambos superiores a 0 e o *identificador* do cliente é uma chave estrangeira que referencia a tabela clientes.

## 5.4 Tabela Reparação

É guardado um *identificador*, na forma de uma chave primária de *auto incremento*, a *designação* de reparação, o seu *custo*, a *data de serviço*, a *data de início* de reparação, a *data final* de reparação e o *identificador* do cliente. Todos os atributos com exceção da *designação* de reparação são opcionais, é verificado que o custo é maior que zero e que a data final de reparação é superior á data de início. O *identificador* de cliente é uma chave estrangeira que referencia a tabela clientes.

## 5.5 Tabela Funcionário

É guardado um *identificador*, na forma de uma chave primária de *auto incremento*, o *nome, morada, email, telemóvel, salário base* e o *identificador* de especialidade. Todos os atributos com exceção do *telemóvel* são obrigatórios. O *identificador* de especialidade é uma chave estrangeira que referencia a tabela especialidade.

## 5.6 Tabela Salário Mensal

É guardado um *identificador*, na forma de uma chave primária de *auto incremento*, o *mês*, o *salário*, o *ano* e o *identificador* do funcionário. Todos os atributos são obrigatórios e é verificado que o *salário* é maior que zero, e se o *mês* está entre 1 e 12. O *identificador* do funcionário é uma chave estrangeira que referencia a tabela funcionário.

5.6 Tabela Incremento Venda

Trata-se de uma tabela de relação onde é guardado o *identificador* da venda e do funcionário. A chave primária é composta por ambos e devido a isto também são atributos não nulos. O *identificador* da venda é uma chave estrangeira que referencia a tabela venda, o *identificador* do funcionário é uma chave estrangeira que referencia a tabela funcionário.

## 5.7 Tabela Incremento Reparação

Trata-se de uma tabela de relação onde é guardado o *identificador* da reparação, do funcionário e também o número de *horas gastas*. Por norma é assumido que o número de *horas gastas* é 0 e é verificado que o número inserido é maior que 0. A chave primária é composta por ambos os *identificadore*s e devido a isto também são atributos não nulos. O *identificador* da reparação é uma chave estrangeira que referencia a tabela reparação, o *identificador* do funcionário é uma chave estrangeira que referencia a tabela funcionário.

## 5.8 Tabela Peça

É guardado um *identificador*, na forma de uma chave primária de *auto incremento*, o *modelo*, o preço de *venda* e o *stock*. Todos os campos são obrigatórios. É ainda verificado que o preço de venda é maior que 0 e o stock é maior ou igual a 0.

## 4.9 Tabela Peça Reparação

Trata-se de uma tabela de relação onde é guardado o *identificador* da peça e da reparação. A chave primária é composta por ambos e devido a isto também são atributos não nulos. O *identificador* da peça é uma chave estrangeira que referencia a tabela peça, o *identificador* da reparação é uma chave estrangeira que referencia a tabela reparação.

## 5.10 Tabela Peça Fornecedor

Trata-se de uma tabela de relação onde é guardado o *identificador* da peça, do fornecedor e o preço de compra. A chave primária é composta por ambos os *identificadores* e devido a isto também são atributos não nulos. O preço de compra também é obrigatório. O *identificador* da peça é uma chave estrangeira que referencia a tabela peça, o *identificador* do fornecedor é uma chave estrangeira que referencia a tabela fornecedor.

## 5.11 Tabela Peça Venda

Trata-se de uma tabela de relação onde é guardado o *identificador* da peça, da venda e o número de peças. A chave primária é composta por ambos os *identificadores* e devido a isto também são atributos não nulos. O número de peças também é obrigatório. O *identificador* da peça é uma chave estrangeira que referencia a tabela peça, o *identificador* da venda é uma chave estrangeira que referencia a tabela venda.

## 5.12 Tabela Fornecedor

É guardado um *identificador*, na forma de uma chave primária de *auto incremento*, e o *nome* do fornecedor. Ambos os campos são obrigatórios.

## 5.13 Tabela Armazém

É guardado um *identificador*, na forma de uma chave primária de *auto incremento*, o *nome, morada, área* e *identificador* do fornecedor. Todos os atributos são obrigatórios e é verificado que a área é maior que 0. O *identificador* do fornecedor é uma chave estrangeira que referencia a tabela fornecedor.