**1**

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](#_Toc90128534)

[2.Diagrama UML 3](#_Toc90128535)

[3.Esquema Relacional 4](#_Toc90128536)

[5.Implementação de restrições na Base de dados 6](#_Toc90128537)

[5.1 Tabela Clientes 6](#_Toc90128538)

[5.2 Tabela Venda 6](#_Toc90128539)

[5.3 Tabela Reparação 6](#_Toc90128540)

[5.5 Tabela Salário Mensal 7](#_Toc90128541)

[5.7 Tabela Incremento Reparação 7](#_Toc90128542)

[5.8 Tabela Peça 8](#_Toc90128543)

[5.9 Tabela Peça Reparação 8](#_Toc90128544)

[5.10 Tabela Peça Fornecedor 8](#_Toc90128545)

[5.11 Tabela Peça Venda 8](#_Toc90128546)

[5.12 Tabela Fornecedor 8](#_Toc90128547)

[5.13 Tabela Armazém 8](#_Toc90128548)

# 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.

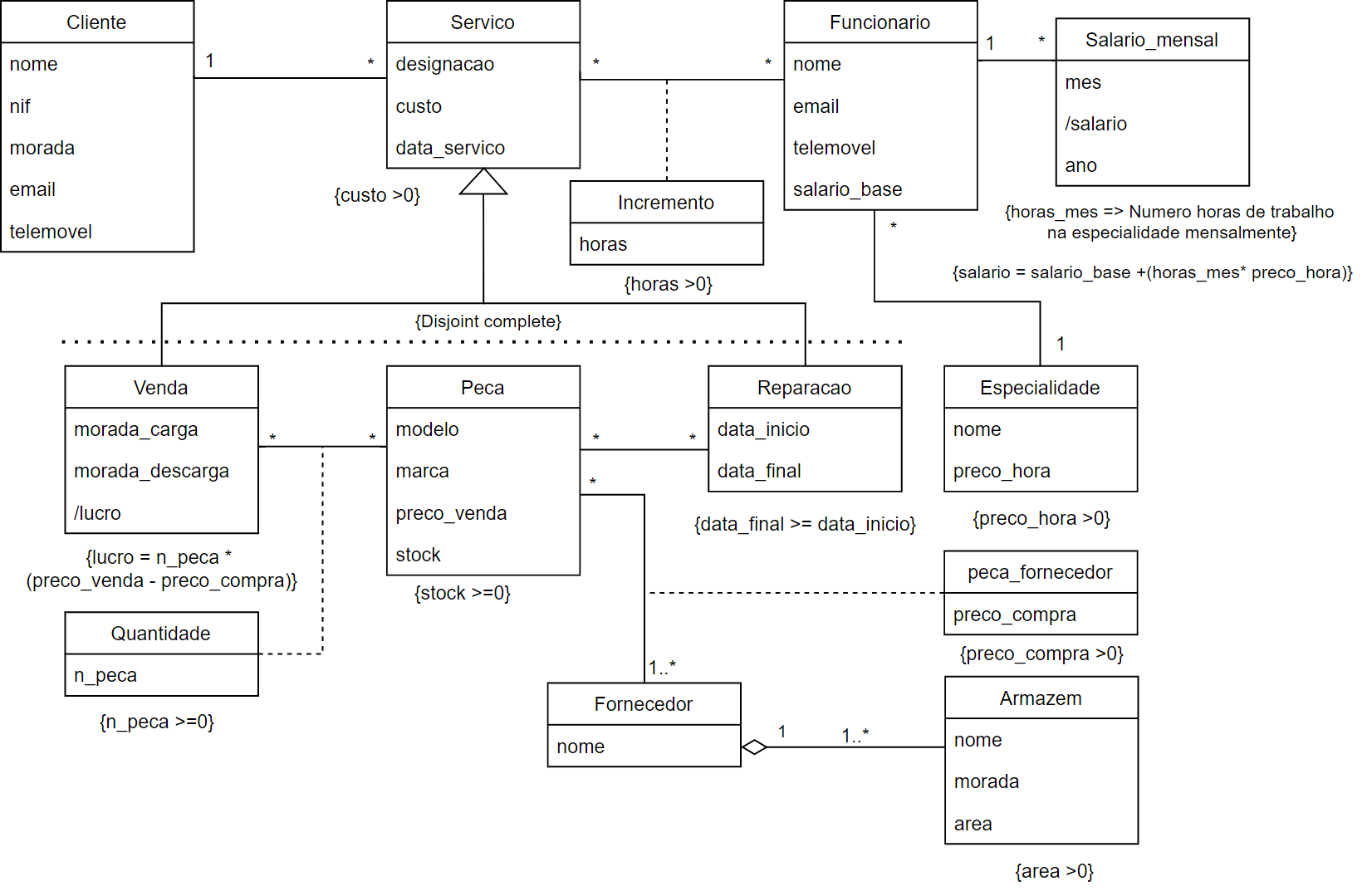
A loja guarda um registo de todas as peças de hardware disponíveis. Estas são utilizadas em ambos os serviços disponibilizados. Sobre cada peça sabe-se o modelo, a marca, a quantidade e o preço a que é vendida ao cliente.

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 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 Reparacao

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 Not Null,
* Telemovel 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.