

Faculdade de Engenharia da Universidade do Porto



Bases de Dados

Licenciatura em Engenharia Informática e Computação

Projeto 21/22



Turma 7 Grupo 6:

Ana Rita Carneiro – up202008569

Milena Gouveia – up202008862

Sofia Moura – up201907201

ÍNDICE

Contexto	3
Modelo Conceptual	4
Modelo conceptual (atualizado)	5
Modelo Relacional	6
Análise de Dependências Funcionais e Formas Normais	7
Restrições na base de dados	8

Contexto

Uma loja de roupa online tem como objetivo gerir o stock e as vendas efetuadas.

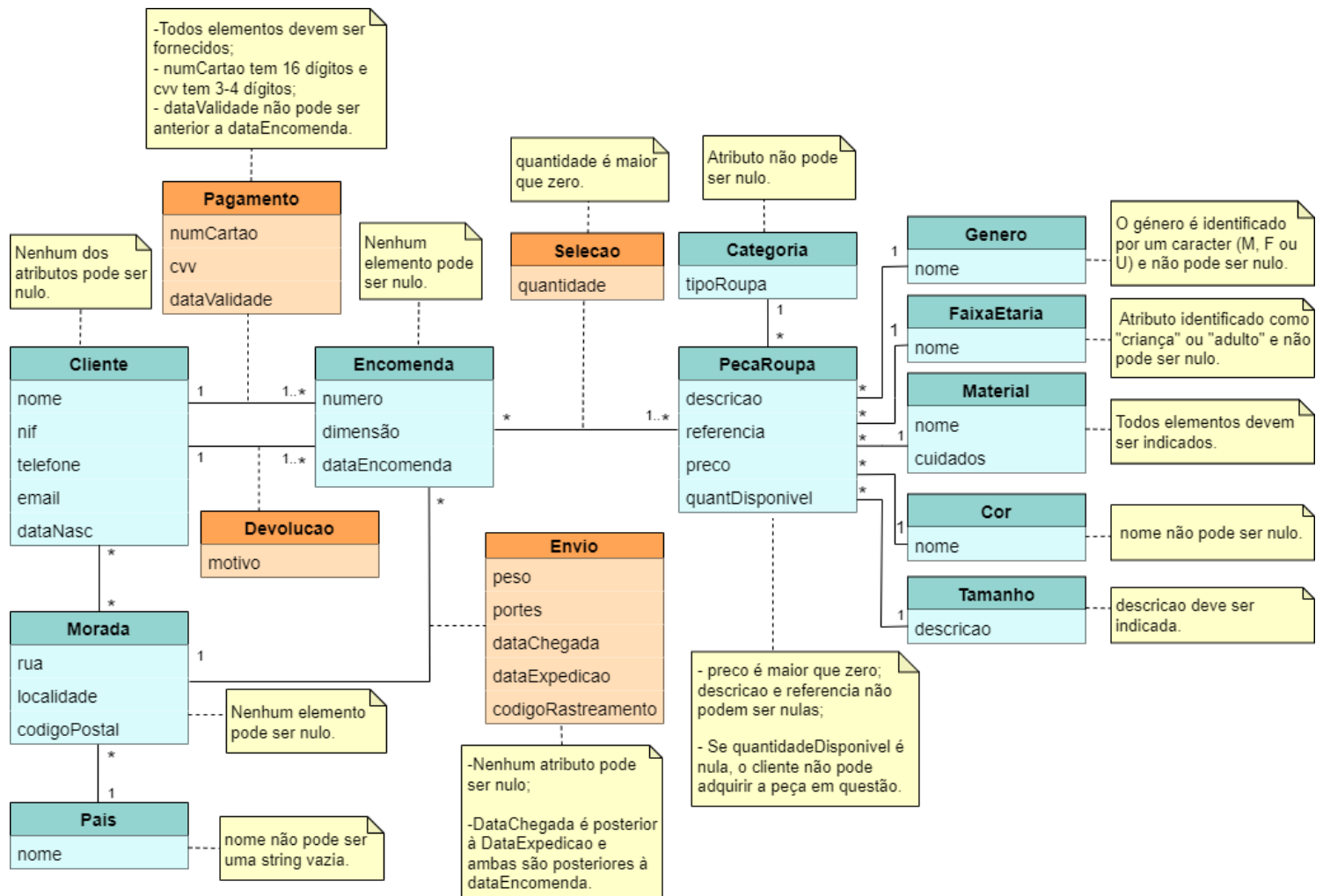
Antes de um cliente efetuar uma compra, tem de criar uma conta fornecendo ao site informações tal como o seu nome, telefone, email, NIF e data de nascimento. Tem também de definir uma ou mais moradas para as quais serão entregues as encomendas que efetuar, indicando o respetivo país, localidade, rua e código postal, selecionando uma delas no final de cada pedido.

A cada encomenda efetuada por um respetivo cliente é atribuído um número que a identifica e registada a sua dimensão total e a data em que foi efetuada. Em relação ao envio da mesma é fornecida informação acerca dos portes de envio, peso do pedido, data de expedição, data prevista de chegada. Em caso de devolução, o cliente deve indicar devidamente o motivo da mesma. Antes de terminar a compra, o cliente seleciona o método de pagamento desejado, indicando o número do cartão, data de validade e CVV.

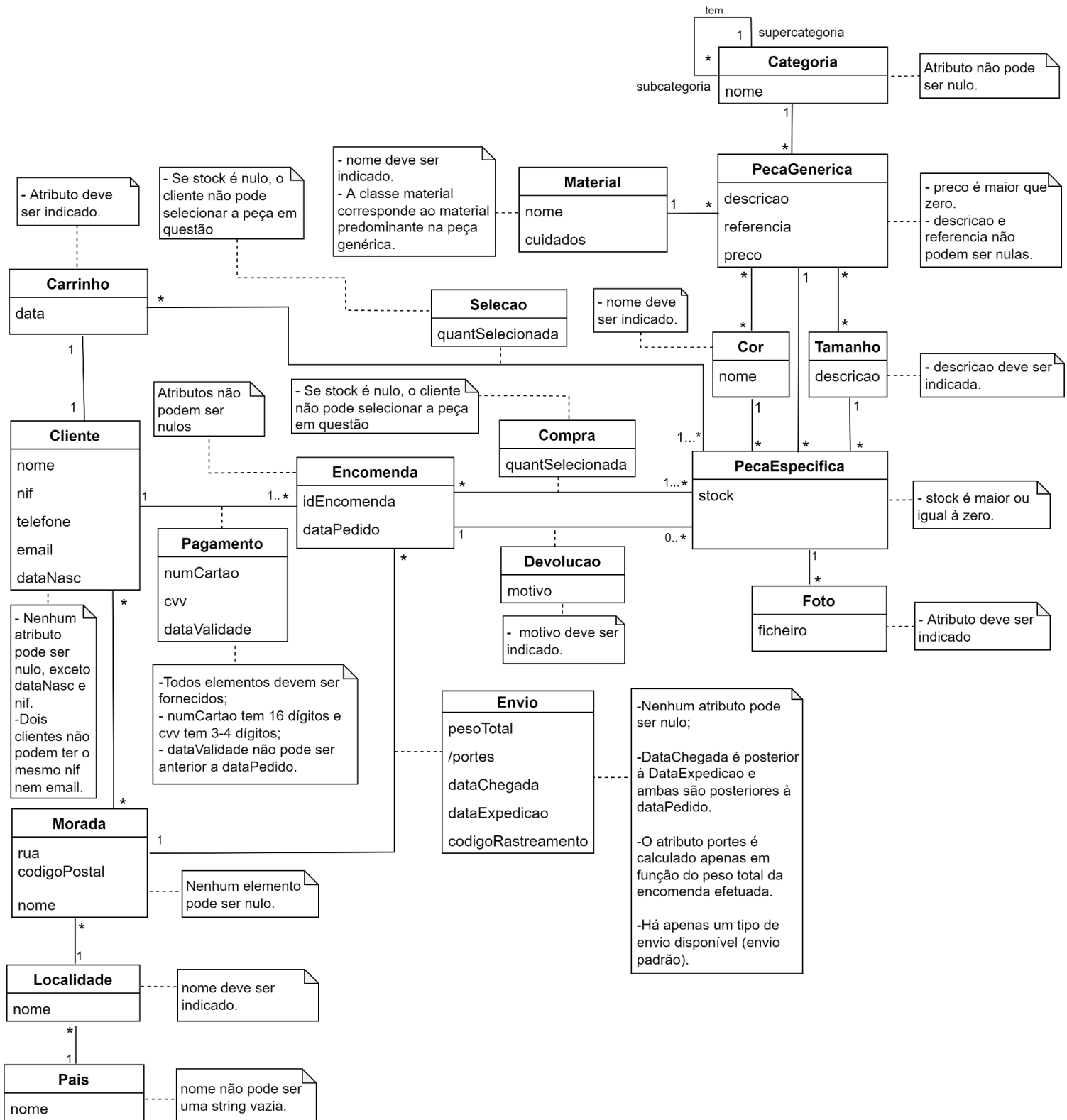
Em cada pedido, o utilizador seleciona as várias peças desejadas e a quantidade de cada uma. Em relação a estas, o site fornece uma pequena descrição de cada uma, uma referência única a cada produto, um preço e ainda o material presente em maior quantidade na peça e os cuidados a ter com o mesmo. Guarda também informação referente ao stock existente de cada peça. Nesta seleção, é escolhido também o tamanho e a cor do item.

A empresa organiza as roupas em diferentes faixas etárias, género e tipo de peça.

Modelo Conceptual



Modelo conceptual (atualizado)



Modelo Relacional

Cliente (idCliente, nome, nif, telefone, email, dataNasc)

Morada (idMorada, nome, rua, codigoPostal, idLocalidade->Localidade)

ClienteMorada (idCliente->Cliente, idMorada->Morada)

Pais (idPais, nome)

Localidade (idLocalidade, nome, idPais->Pais)

Carrinho (idCarrinho, dataCriacao, idCliente->Cliente)

Encomenda (idEncomenda, dataPedido)

Pagamento (idCliente->Cliente, idEncomenda->Encomenda, numCartao, cvv, dataValidade)

Envio (idEncomenda->Encomenda, idMorada->Morada, pesoTotal, portes, dataExpedicao, dataChegada, codigoRastreamento)

PecaEspecifica (idPecaEspecifica, stock, idCor->Cor, idTamanho->Tamanho, idPecaGenerica->PecaGenerica)

Devolucao (idEncomenda->Encomenda, idPecaEspecifica->PecaEspecifica, motivo)

Compra (idEncomenda->Encomenda, idPecaEspecifica->PecaEspecifica, quantSelecionada)

Selecao (idCarrinho->Carrinho, idPecaEspecifica->PecaEspecifica, quantSelecionada)

Foto (idFoto, ficheiro, idPecaEspecifica->PecaEspecifica)

PecaGenerica (idPecaGenerica, descricao, referencia, preco, idMaterial->Material, idCategoria->Categoria)

Categoria (idCategoria, nome, superCategoria-> Categoria)

Material (idMaterial, nome, cuidados)

Cor (idCor, nome)

PecaGenericaCor (idPecaGenerica->PecaGenerica, idCor->Cor)

Tamanho (idTamanho, descricao)

PecaGenericaTamanho (idPecaGenerica->PecaGenerica, idTamanho->Tamanho)

Análise de Dependências Funcionais e Formas Normais

Dependências funcionais:

Cliente: {idCliente} -> {nome, nif, telefone, email, dataNasc}

{email} -> {idCliente, nome, nif, telefone, dataNasc}

Morada: {idMorada} -> {nome, rua, codigoPostal, idLocalidade}

{nome, rua, codigoPostal, idLocalidade} -> {idMorada}

Pais: {idPais} -> {nome}

{nome} -> {idPais}

Localidade: {idLocalidade} -> {nome, idPais}

{nome, idPais} -> {idLocalidade}

Carrinho: {idCarrinho} -> {dataCriacao, idCliente}

{dataCriacao, idCliente} -> {idCarrinho}

Encomenda: {idEncomenda} -> {dataPedido}

Peca Especifica: {idPecaEspecificas} -> {stock, idCor, idTamanho, idPecaGenerica}

{idCor, idTamanho, idPecaGenerica} -> {idPecaEspecificas, stock}

Foto: {idFoto} -> {ficheiro, idPecaEspecificas}

{ficheiro, idPecaEspecificas} -> {idFoto}

Peca Generica: {idPecaGenerica} -> {descricao, referencia, preco, idMaterial, idCategoria}

{referencia} -> {idPecaGenerica, descricao, preco, idMaterial, idCategoria }

Categoria: {idCategoria} -> {nome}

{nome} -> {idCategoria}

Material: {idMaterial} -> {nome, cuidados}

{nome} -> {idMaterial, cuidados}

Cor: {idCor} -> {nome}

{nome} -> {idCor}

Tamanho: {idTamanho} -> {descricao}

{descricao} -> {idTamanho}

Pagamento: {idCliente, idEncomenda} -> {numCartao, cvv, dataValidade}

Envio: {idEncomenda} -> {idMorada, pesoTotal, portes, dataChegada, dataExpedicao, codigoRastreamento}

{pesoTotal} -> {portes}

```
{codigoRastreamento} -> {idMorada, pesoTotal, portes, dataChegada, dataExpedicao}
```

Formas normais:

Forma normal de Boyce-Codd (BCNF):

{pesoTotal} -> {portes} - viola a BCNF, uma vez que a dependência não é trivial e o atributo pesoTotal não é uma chave.

3ª Forma normal (3NF):

{pesoTotal} -> {portes} - sendo não trivial, viola a 3FN, uma vez que, o atributo pesoTotal não é uma chave e do lado direito da dependência não existem atributos primos.

Decomposição para a BCNF:

{pesoTotal} -> {portes}

Decomposição da relação Envio:

Envio1(pesoTotal, portes)

```
Envio2(pesoTotal, idEncomenda, idMorada, dataChegada, dataExpedicao,
codigoRastreamento)
```

Determinação das formas normais e chaves de Envio1:

{pesoTotal} -> {portes}

chave - {pesoTotal} Está na BCNF

Determinar formas normais e chaves de Envio2:

```
{idEncomenda} -> {pesoTotal, idMorada, dataChegada, dataExpedicao,
codigoRastreamento}
```

chave - {idEncomenda} Está na BCNF

Todas as restantes relações seguem a Forma Normal de Boyce-Codd e a 3ª Forma Normal uma vez que, não sendo triviais, a parte esquerda de cada uma permite chegar a todos os restantes atributos da mesma, sendo por isso, uma chave.

Restrições na base de dados

Cliente:

Não podem existir dois clientes com o mesmo id: **idCliente PRIMARY KEY**

Atributos nome, telefone e email devem ser indicados: **nome NOT NULL, telefone NOT NULL, email NOT NULL**

Dois clientes não podem ter o mesmo nif nem o mesmo email: **email UNIQUE, nif UNIQUE**

Morada:

Não podem existir duas moradas com o mesmo id: **idMorada PRIMARY KEY**

Atributos nome, rua, codigoPostal devem ser indicados: **nome NOT NULL, rua NOT NULL, codigoPostal NOT NULL**

O id da localidade deve corresponder a um id de uma instância da tabela Localidade: **idLocalidade REFERENCES Localidade(idLocalidade)** e deve ser indicado: **idLocalidade NOT NULL**

Deve existir apenas uma morada para cada combinação diferente de nome, rua, codigoPostal e idPais: **UNIQUE(nome, rua, codigoPostal, idLocalidade)**

ClienteMorada:

O id do cliente e da morada deve corresponder a um id de uma instância das tabelas Cliente e Morada respetivamente: **idCliente REFERENCES Cliente (idCliente), idMorada REFERENCES Morada (idMorada)**

O id do cliente e da morada não deve ser nulo: **idCliente NOT NULL, idMorada NOT NULL**

Pais:

Não podem existir dois países com o mesmo id: **idPais PRIMARY KEY**

O nome do país não deve ser nulo: **nome NOT NULL**

Não podem existir dois países com o mesmo nome: **nome UNIQUE**

Localidade:

Duas localidades não podem ter o mesmo número de identificação: **idLocalidade PRIMARY KEY**

É obrigatória a indicação do nome da localidade: **nome NOT NULL** e este deve ser único **nome UNIQUE**

O id do país deve corresponder a uma instância da tabela Pais: **idPais REFERENCES Pais(idPais)**

Carrinho:

Não podem existir dois carrinhos com o mesmo id: **idCarrinho PRIMARY KEY**

A data de criação do carrinho não pode ser nula: **dataCriacao NOT NULL**

O id do cliente deve corresponder a um id de uma instância da tabela Cliente: **idCliente REFERENCES Cliente (idCliente)**

O id do cliente não deve ser nulo: **idCliente NOT NULL**

Deve existir apenas um carrinho para cada combinação diferente de data de criação e id do cliente: **UNIQUE (dataCriacao, idCliente)**

Encomenda:

Todas encomendas devem ter id diferente: **idEncomenda PRIMARY KEY**

A data de pedido deve ser indicada: **dataPedido NOT NULL**

Pagamento:

O id do cliente e da encomenda deve corresponder a um id de uma instância da tabela Cliente e Encomenda, respetivamente: **idCliente REFERENCES Cliente (idCliente), idEncomenda REFERENCES Encomenda (idEncomenda)** e devem ser indicados: **idCliente NOT NULL, idEncomenda NOT NULL**

Atributos numCartao, cvv e dataValidade devem ser indicados: **numCartao NOT NULL, cvv NOT NULL, dataValidade NOT NULL**

Envio1:

O peso total do envio e os seus portes devem ser indicados: **pesoTotal UNIQUE, portes UNIQUE**

O valor do peso total deve ser superior a 0: **pesoTotal CHECK(pesoTotal>=0)**

Envio2:

O id da encomenda e da morada deve corresponder a um id de uma instância da tabela Encomenda e Morada, respetivamente: **idEncomenda REFERENCES Encomenda (idEncomenda), idMorada REFERENCES Morada (idMorada)** e não podem existir encomendas nem moradas com o mesmo id: **idEncomenda PRIMARY KEY, idMorada PRIMARY KEY**

As datas de chegada e de expedição e o código de rastreamento devem ser indicados: **dataChegada NOT NULL, dataExpedicao NOT NULL, codigoRastreamento NOT NULL**

Não pode existir mais que uma encomenda com o mesmo código de rastreamento: **codigoRastreamento UNIQUE**

PecaEspecific:

Duas peças não podem ter o mesmo id: **idPecaEspecific PRIMARY KEY**

O stock deve ser maior ou igual a zero: **CHECK (stock>=0)**

o id da cor, do tamanho e da peça genérica deve corresponder a uma instância da tabela Cor, Tamanho e PecaGenerica e além disso não devem ser nulos: **idCor NOT NULL REFERENCES Cor (idCor), idTamanho NOT NULL REFERENCES TAMANHO (idTamanho), idPecaGenerica NOT NULL REFERENCES PecaEspecific (idPecaEspecific)**

Deve existir apenas uma peça específica para a mesma combinação de cor, tamanho, pecaGenerica: **UNIQUE (idCor, idTamanho, idPecaGenerica)**

Devolucao:

O id da encomenda e da peça específica deve corresponder a um id de uma instância das tabelas Encomenda e PecaEspecific respetivamente: **idEncomenda REFERENCES Encomenda(idEncomenda), idPecaEspecific REFERENCES PecaEspecific(idPecaEspecific)**

O id da encomenda não deve ser nulo: **idEncomenda NOT NULL**

Não podem haver duas peças específicas com o mesmo id: **idPecaEspecific PRIMARY KEY**

O motivo da devolução não deve ser nulo: **motivo NOT NULL**

Compra:

O id da encomenda e da peça específica devem corresponder a um id de uma instância da tabela Encomenda e PecaEspecific, respetivamente: **idEncomenda REFERENCES Encomenda(idEncomenda), idPecaEspecific REFERENCES PecaEspecific(idPecaEspecific)** e devem ser indicados: **idEncomenda NOT NULL, idPecaEspecific NOT NULL**

Não deve haver duas instâncias com o mesmo par (idEncomenda,idPecaEspecifica):
PRIMARY KEY (idEncomenda,idPecaEspecifica)

Selecao:

O id do cliente e da peça específica devem corresponder a um id de uma instância da tabela Cliente e PecaEspecifica, respetivamente: **idCliente REFERENCES**

Cliente(idCliente), idPecaEspecifica REFERENCES

PecaEspecifica(idPecaEspecifica) e devem ser indicados: **idCliente NOT NULL, idPecaEspecifica NOT NULL**

Não deve haver duas instâncias com o mesmo par (idEncomenda,idPecaEspecifica):
PRIMARY KEY (idEncomenda,idPecaEspecifica)

Foto:

Não podem existir duas fotos com o mesmo id: **idFoto PRIMARY KEY**

Os atributos ficheiro e idPecaEspecifica devem ser indicados: **ficheiro NOT NULL, idPecaEspecifica NOT NULL**

O id da peça específica deve corresponder a um id de uma instância da tabela

Localidade: **idPecaEspecifica REFERENCES PecaEspecifica(idPecaEspecifica)**

Deve existir apenas uma foto para cada combinação diferente de ficheiro e

idPecaEspecifica: **UNIQUE(ficheiro, idPecaEspecifica)**

PecaGenerica:

Cada peça genérica tem o seu id diferente de todas as outras peças: **idPecaGenerica PRIMARY KEY**

A descrição, a referência, o preço, o material e a categoria sempre devem ser indicadas: **descricao NOT NULL, referencia NOT NULL, preco NOT NULL, idMaterial NOT NULL, idCategoria NOT NULL**

Além disso, a descrição é única para cada peça: **descricao UNIQUE**

O id do material e da categoria correspondem a uma instância das tabelas Material e Categoria respetivamente: **idMaterial REFERENCES Material(idMaterial), idCategoria REFERENCES Categoria(idCategoria)**

Categoria:

Não podem existir duas categorias com o mesmo id: **idCategoria PRIMARY KEY**

Os atributos nome e superCategoria devem ser indicados: **nome NOT NULL, superCategoria NOT NULL** e devem ser únicos: **nome UNIQUE, superCategoria UNIQUE**

Material:

Cada material possui uma identificação diferente de todos os outros: **idCor PRIMARY KEY**

A indicação do nome do material é obrigatória: **nome NOT NULL**

Cor:

O id de duas cores deve ser sempre diferente: **idCor PRIMARY KEY**

O nome deve ser indicado: **nome NOT NULL** e não devem existir duas cores com o mesmo nome: **nome UNIQUE**

PecaGenericaCor:

O id da peça genérica e o id da cor devem corresponder a uma instância das tabelas PecaGenerica e Cor respetivamente: **idPecaGenerica REFERENCES**

PecaGenerica(idPecaGenerica), idCor REFERENCES Cor(idCor)

Nem o id da peça genérica nem o id da cor devem ser nulos: **idPecaGenerica NOT NULL, idCor NOT NULL** e ambos devem ser únicos: **idPecaGenerica PRIMARY KEY, idCor PRIMARY KEY**

Tamanho:

Não podem existir dois tamanhos com o mesmo id: **idTamanho PRIMARY KEY**
O atributo descricao deve ser indicado: **descricao NOT NULL**, cada tamanho tem uma descrição diferente: **descricao UNIQUE** e o valor da descrição deve ser '32', '34', '36', '38', '40', '42', '44', 'XXS', 'XS', 'S', 'M', 'L', 'XL' ou 'XXL'

PecaGenericaTamanho:

O id da peça genérica e do tamanho deve corresponder a uma instância das tabelas PecaGenerica e Tamanho respetivamente: **idPecaGenerica REFERENCES PecaGenerica(idPecaGenerica), idTamanho REFERENCES Tamanho(idTamanho)**
O id da peça genérica e o id do tamanho não devem ser nulos: **idPecaGenerica NOT NULL, idTamanho NOT NULL** e devem ser únicos: **idPecaGenerica PRIMARY KEY, idTamanho PRIMARY KEY**