



U N I V E R S I D A D E  
**LUSÓFONA**  
D O P O R T O

---

## *RELATÓRIO BASE DE DADOS*

---

**Gabriel Faria de Souza Lima - 21800381**

**Universidade Lusófona do Porto**

**Base de dados**

dezembro 2019

## **SUMÁRIO**

|  |          |
|--|----------|
| <b>Introdução.....</b>                 | <b>3</b> |
| <b>Objetivos da Base de Dados.....</b> | <b>3</b> |
| <b>Modelo Entidade-Associação.....</b> | <b>4</b> |
| <b>Modelo Entidade-Associação.....</b> | <b>5</b> |
| <b>Modelo Relacional.....</b>          | <b>6</b> |
| <b>Normalização.....</b>               | <b>7</b> |
| <b>SQL - LDD.....</b>                  | <b>8</b> |
| <b>SQL - LMD.....</b>                  | <b>9</b> |
| <b>Conclusões.....</b>                 | <b>9</b> |

# INTRODUÇÃO

- Tema escolhido:

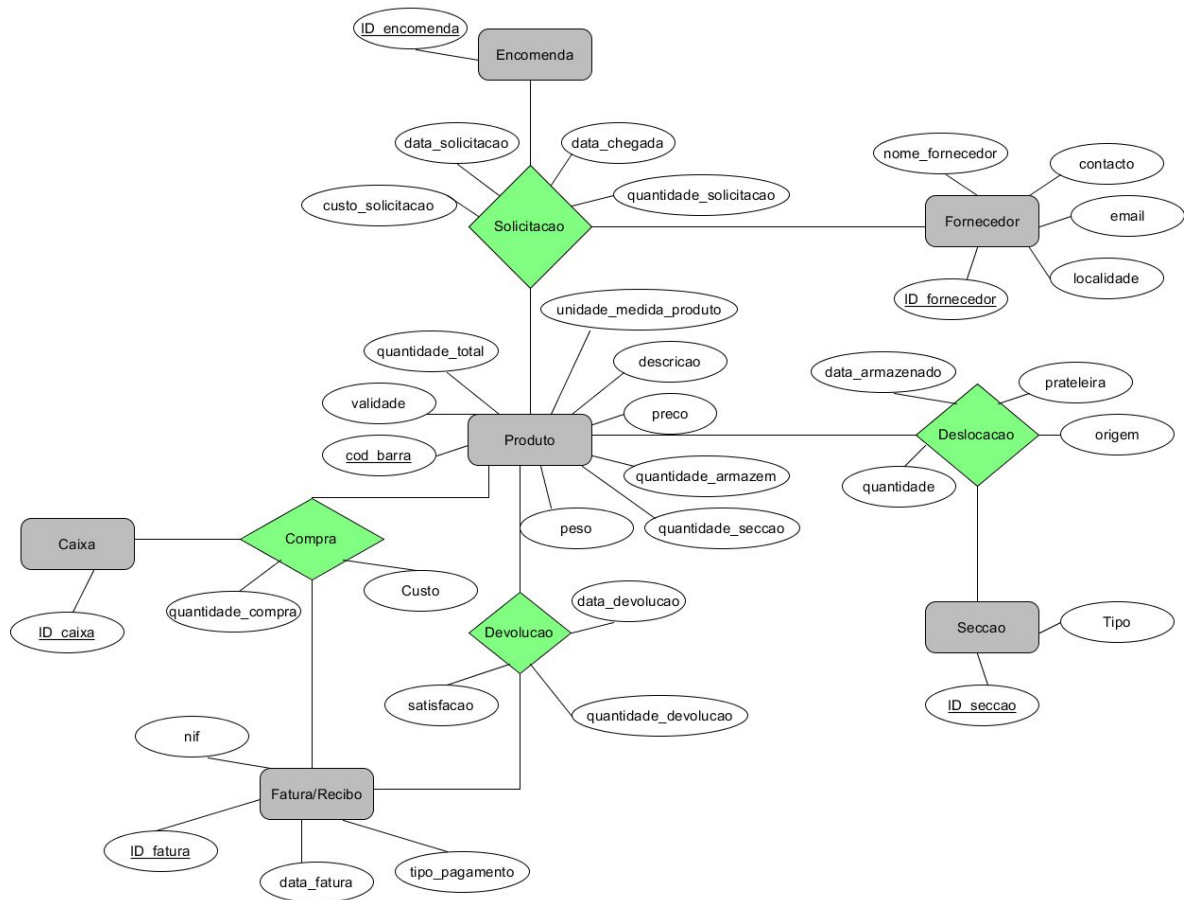
Gestão de produtos de um hipermercado.

- Objetivos da base de dados:

A base de dados foi desenvolvida tendo em mente todas as operações de um produto de hipermercado, como a sua venda e o respectivo caixa, sua solicitação por encomendas, o processo de deslocação entre armazém, secções e prateleiras, e permitir a devolução de um item caso o cliente não estiver satisfeito.

O maior objetivo da base de dados é armazenar todo tipo de informação que possa estar envolvida nessas operações e movimentações do produto, facilitando o processo de registro de todos os módulos que se espera em um ambiente de hipermercado. Com esta base de dados é possível tirar todo esse tipo de informação na integração da mesma no projeto conjunto.

# MODELO ENTIDADE-ASSOCIAÇÃO



## ● Contextualização:

Legenda: <Nome de uma associação>,[Nome de uma entidade].

A figura mostra que a entidade principal desse modelo será o produto.

Entidade esta que participa efetivamente com todas as outras através de associações. O produto tem uma associação <Solicitação> diretamente relacionada com encomendas e fornecedor que por possuir atributos próprios dá origem a uma tabela com uma chave sendo composta pelas as chaves das entidades associadas (produto,encomenda e fornecedor),

todas as entidades associadas também geram uma tabela e possuem seus atributos que são as características que permitem descrever as entidades e associações.

As associações foram escolhidas após algumas abordagens de qual seria a melhor forma de registrar o processo de movimentação do produto e a decisão mais pertinente foi a de cada uma dessas associações terem seus atributos próprios, nesse modelo podemos ver <Compra>, <Devolução>, <Deslocação>, <Solicitação>.

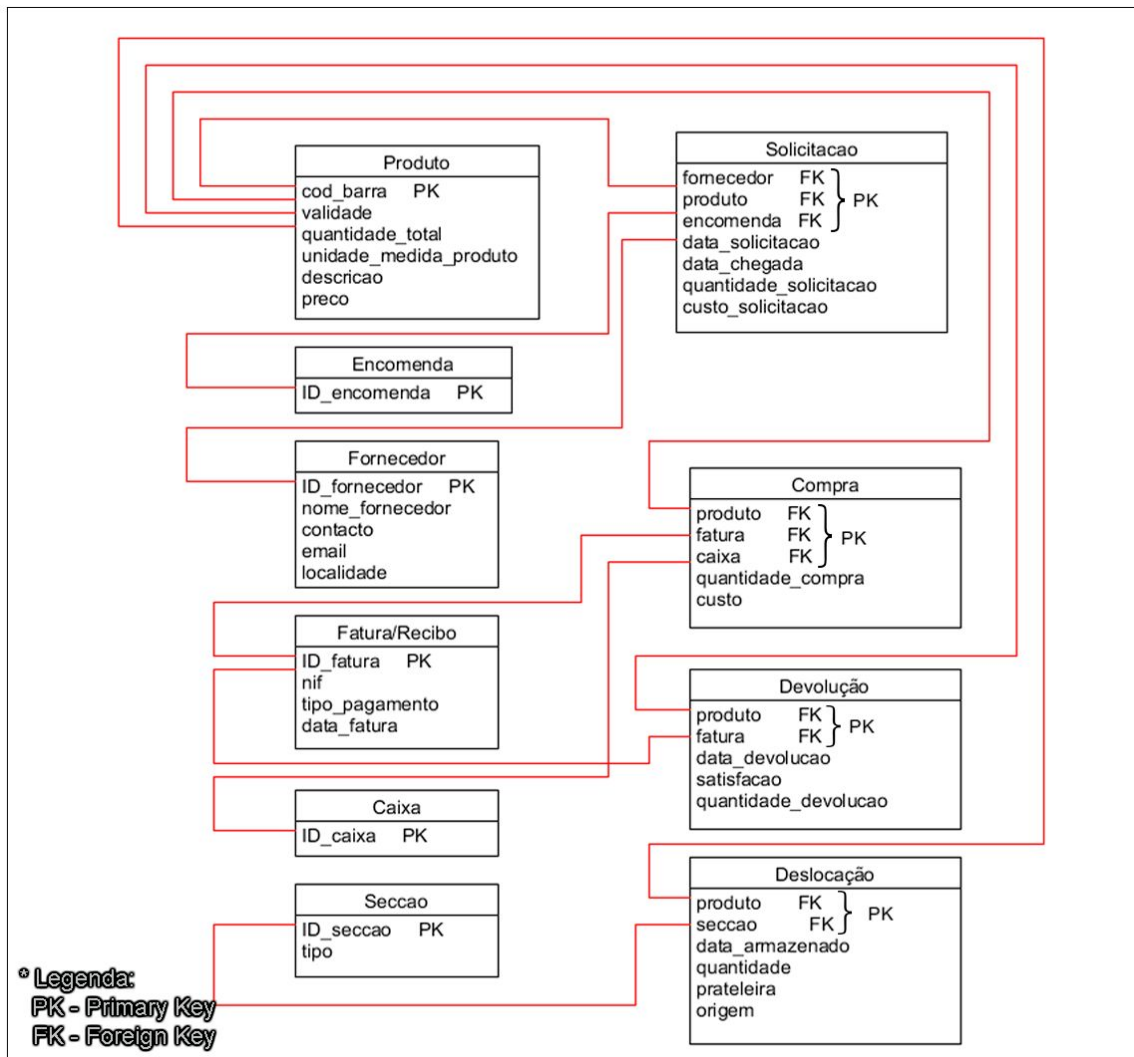
<Compra> tem seus atributos próprios armazenando a quantidade da compra, o custo desta e é diretamente relacionada com [Produto] ,[Caixa] e [Fatura/recibo] tendo uma chave composta que referencia qual produto foi comprado, em que caixa foi comprado e a qual fatura essa compra está associada.

Em caso de insatisfação do cliente, <Devolução> tem seus atributos próprios armazenando a quantidade da devolução, qual foi o motivo da devolução do produto, ou seja a satisfação do cliente e a data dessa devolução, é diretamente relacionada com [Produto] e [Fatura/recibo] tendo uma chave composta que referencia qual produto foi devolvido e a qual fatura essa devolução está associada.

<Deslocação> tem seus atributos próprios que giram em torno de produto armazenando a quantidade armazenada, a prateleira que está armazenado e a origem desse produto ou seja em que secção ele estava antes de estar na secção atualmente armazenado, é diretamente relacionada com [Produto] ,[Secção] tendo uma chave composta que referencia qual produto está sendo deslocado, e em qual secção este produto está sendo armazenado.

<Solicitação> tem seus atributos próprios que giram em torno de produto armazenando a data da solicitação e a data de chegada da encomenda do produto, a quantidade solicitada e o custo desta solicitação, é diretamente relacionada com [Produto] ,[Encomenda] e [Fornecedor] tendo uma chave composta que referencia qual produto está sendo encomendado, de qual fornecedor pertence esse produto e em qual encomenda essa solicitação está associada.

# MODELO RELACIONAL



## ● Contextualização:

O modelo relacional tem como objetivo demonstrar de uma forma mais natural todas as tabelas geradas pelo modelo entidade associação (pagina 3), demonstrando as associações entre cada tabela e demonstrando como as chaves (valores da coluna PK) são referenciadas (FK)

## ● Normalização:

O método de normalização surge como forma de transformar a base de dados em uma relação arbitrária, com suas respectivas restrições e características.

### 1FN:

A primeira forma normal implica que nenhum atributo pode admitir um conjunto de valores, como por exemplo um produto ter duas descrições.

A base de dados desse documento está normalizada na primeira forma normal.

### 2FN:

A segunda forma normal exige que todas as tabelas não possuam dependências parciais. Ou seja, em uma tabela que contenha duas chaves ou mais (chave composta) possuírem atributos que necessitam apenas de um elemento da chave e não a chave composta, como por exemplo se nossa tabela “solicitação” possuíse uma coluna “endereço fornecedor” que apenas dependeria da chave do fornecedor e não a chave composta entre fornecedor, produto e encomenda.

A base de dados desse documento está normalizada na segunda forma normal.

### 3FN:

A terceira forma normal exige que todas as tabelas não possuam dependências parciais visto na 2FN e nem dependências transitivas, ou seja os atributos têm de ser dependentes apenas do atributo chave primária (PK), não podendo ser dependentes de outro atributo dentro da mesma tabela, como por exemplo se dentro de fatura tivéssemos o código do produto juntamente com sua descrição, que seria dependente apenas do código do produto e não da chave primária id\_fatura.

A base de dados desse documento está normalizada na terceira forma normal.

# SQL - LDD

```
create database hipermercado_market_management

CREATE TABLE produto(
  cod_barra int PRIMARY KEY NOT NULL,
  quantidade_total int DEFAULT NULL,
  quantidade_armazem int DEFAULT NULL,
  quantidade_seccao int DEFAULT NULL,
  descricao varchar(50) NOT NULL,
  preco_unitario float NOT NULL,
  peso varchar(10) NOT NULL,
  unidade_medida_produto varchar(20) NOT NULL,
  validade date DEFAULT NULL
);

CREATE TABLE fatura(
  ID_fatura int PRIMARY KEY NOT NULL,
  nif int DEFAULT NULL,
  data_fatura date NOT NULL DEFAULT current_timestamp,
  tipo_pagamento varchar(50) NOT NULL
);

CREATE TABLE caixa (
  ID_caixa int PRIMARY KEY NOT NULL
);

CREATE TABLE seccao (
  ID_seccao int PRIMARY KEY NOT NULL,
  tipo varchar(20) NOT NULL
);

CREATE TABLE fornecedor (
  ID_fornecedor int PRIMARY KEY NOT NULL,
  nome_fornecedor varchar NOT NULL,
  contacto varchar(15) NOT NULL,
  email varchar(30) NOT NULL,
  localidade varchar(20) NOT NULL
);

CREATE TABLE encomenda (
  ID_encomenda int PRIMARY KEY NOT NULL
);

CREATE TABLE compra (
  caixa_fk int foreign key(caixa_fk) references caixa(ID_caixa) NOT NULL,
  fatura_fk int foreign key(fatura_fk) references fatura(ID_fatura) NOT NULL,
  produto_fk int foreign key(produto_fk) references produto(cod_barra) NOT NULL,
  PRIMARY KEY(caixa_fk, fatura_fk, produto_fk),
  quantidade_compra int NOT NULL,
  custo float DEFAULT NULL
);

CREATE TABLE devolucao (
  produto_fk int foreign key(produto_fk) references produto(cod_barra) NOT NULL,
  fatura_fk int foreign key(fatura_fk) references fatura(ID_fatura) NOT NULL,
  PRIMARY KEY ( produto_fk, fatura_fk),
  data_devolucao date NOT NULL,
  quantidade_devolucao int NOT NULL,
  satisfacao varchar(200) NOT NULL
);

CREATE TABLE deslocação (
  produto_fk int foreign key(produto_fk) references produto(cod_barra) NOT NULL,
  seccao_fk int foreign key(seccao_fk) references seccao(ID_seccao) NOT NULL,
  PRIMARY KEY (produto_fk, seccao_fk),
  prateleira int DEFAULT NULL,
  origem varchar(20) NOT NULL,
  data_deslocação date NOT NULL DEFAULT current_timestamp,
  quantidade_deslocação int NOT NULL
);

CREATE TABLE solicitacao (
  encomenda_fk int foreign key(encomenda_fk) references encomenda(ID_encomenda) NOT NULL,
  produto_fk int foreign key(produto_fk) references produto(cod_barra) NOT NULL,
  fornecedor_fk int foreign key(fornecedor_fk) references fornecedor(ID_fornecedor) NOT NULL,
  PRIMARY KEY (encomenda_fk, produto_fk, fornecedor_fk),
  data_encomenda date NOT NULL DEFAULT current_timestamp,
  data_chegada date NULL DEFAULT NULL,
  quantidade_solicitacao int NOT NULL,
  custo_solicitacao float NOT NULL
);
```

## ● Contextualização:

Criação da base de dados em código SQL, seguindo o modelo relacional, podemos ver todos os atributos assim como suas limitações, de tamanho, tipo da coluna podendo ser uma coluna vazia ou carregada com algum valor de forma padrão como é o caso de algumas datas.

Todas as colunas que possuem varchar tiveram seus tamanhos testados e para desempenhar suas funções o tamanho está de acordo, o ideal sempre é ter o menor tamanho em busca da máxima eficiência da relação.

Algumas colunas têm seus valores em NULL evidenciando que a inserção do valor naquela coluna será feito posteriormente, como por exemplo uma coluna de “data chegada”, “validade” ou cálculo “custo” que será dependente inteiramente do preço que se pede pelo peso do produto. Isso não significa que as colunas irão ter um valor vazio, e sim que elas serão preenchidas utilizando métodos externos ao SQL para o cálculo de seus valores.

As referências de primary key podem ser observadas nas tabelas das associações.



# SQL - LMD

```
INSERT INTO produto (cod_barra, quantidade_total, quantidade_armazen, quantidade_saccao, descricao, preco_unitario, peso, unidade_medida_produto, validade) VALUES
(1, NULL, NULL, NULL, 'Banana', 0.78, '128', 'gr', '2019-11-27'),
(2, NULL, NULL, NULL, 'Maça', 0.75, '780', 'gr', '2019-11-27'),
(3, NULL, NULL, NULL, 'Laranja', 1.05, '250', 'ml', '2019-12-29'),
(4, NULL, NULL, NULL, 'Leite meio gordo', 0.51, '1', 'l', '2019-12-27'),
(5, NULL, NULL, NULL, 'Yoga', 0.3, '50', 'gr', '2019-11-27'),
(7, NULL, NULL, NULL, 'Trampo', 3.8, '1', 'kg', '2019-12-27'),
(8, NULL, NULL, NULL, 'Carne de porco', 4.2, '1', 'kg', '2019-12-24'),
(9, NULL, NULL, NULL, 'Melancia', 3.46, '1', 'kg', '2019-12-27'),
(10, NULL, NULL, NULL, 'Morango', 1.2, '1', 'gr', '2019-12-24'),
(11, NULL, NULL, NULL, 'Queijo', 2, '500', 'gr', '2019-12-25'),
(12, NULL, NULL, NULL, 'Suco em pó', 5, '500', 'gr', '2019-12-28'),
(13, NULL, NULL, NULL, 'Linguiça', 8, '1', 'kg', '2019-12-25'),
(14, NULL, NULL, NULL, 'Lamela', 0.99, '1000', 'gr', NULL);

INSERT INTO fatura (ID_fatura, nif, data_fatura, tipo_pagamento) VALUES
(1, 123456789, '2019-12-31 00:15:18', 'Cartão'),
(2, NULL, '2019-12-31 00:50:28', 'Dinheiro');

INSERT INTO caixa (ID_caixa) VALUES
(1),
(2),
(3),
(4);

INSERT INTO seccao (ID_seccao, tipo) VALUES
(1, 'Armadaz'),
(2, 'Frutas'),
(3, 'Laticínios'),
(4, 'Carne'),
(5, 'Cereais'),
(6, 'Verduras'),
(7, 'Bebidas');

INSERT INTO fornecedor (ID_fornecedor, nome_fornecedor, contacto, email, localidade) VALUES
(1, 'Fast', '123-456-789', 'fast@gmail.com', 'Porto'),
(2, 'Terra', '123-456-789', 'terra@gmail.com', 'Lisboa'),
(3, 'Sol Branco', '123-456-789', 'solbranco@gmail.com', 'Lisboa'),
(4, 'Da Terra', '123-456-789', 'daterra@gmail.com', 'Porto'),
(5, 'Sabon Portugal', '123-456-789', 'sabonportugal@gmail.com', 'Aveiro'),
(6, 'TesteTragas', '789-456-123', 'teste@gmail.com', 'Aveiro');

INSERT INTO encomenda (ID_encomenda) VALUES
(1),
(2),
(3),
(4),
(5);

INSERT INTO compra (caixa_fk, fatura_fk, produto_fk, quantidade_compra, custo) VALUES
(1, 1, 500, NULL),
(1, 1, 3, 1000, NULL),
(1, 1, 7, 30, NULL),
(4, 2, 5, 4500, NULL),
(4, 2, 14, 3000, NULL);

INSERT INTO devolucao (produto_fk, fatura_fk, data_devolucao, quantidade_devolucao, satisfacao) VALUES
(3, 2, '2019-12-31', 300, 'ml'),
(5, 2, '2019-12-31', 500, 'Uva Podre');

INSERT INTO deslocação (produto_fk, seccao_fk, prateleira, origem, data_deslocação, quantidade_deslocação) VALUES
(1, 1, NULL, 'Encomenda', '2019-12-31 17:29:43', 9540),
(2, 1, NULL, 'Encomenda', '2019-12-31 23:17:13', 607),
(2, 2, NULL, 'Encomenda', '2019-12-28 22:46:24', 2369),
(3, 1, NULL, 'Encomenda', '2019-12-29 18:37:23', 6290),
(3, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 550),
(4, 1, NULL, 'Encomenda', '2019-12-31 23:17:13', 40),
(4, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 50),
(5, 1, NULL, 'Encomenda', '2019-12-28 22:46:24', 1500),
(5, 2, NULL, 'Encomenda', '2019-12-28 22:46:24', 3005),
(7, 4, NULL, 'Encomenda', '2019-12-31 17:32:04', 150),
(8, 4, NULL, 'Encomenda', '2019-12-28 22:46:24', 50),
(9, 4, NULL, 'Encomenda', '2019-12-28 22:46:24', 50),
(10, 1, NULL, 'Encomenda', '2019-12-28 22:46:24', 2000),
(10, 2, NULL, 'Encomenda', '2019-12-28 22:46:24', 6050),
(11, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 500),
(12, 5, NULL, 'Encomenda', '2019-12-28 22:46:24', 300),
(13, 4, NULL, 'Armadaz', '2019-12-31 23:12:59', 10),
(14, 1, NULL, 'encomenda', '2019-12-31 17:29:43', 4500);

INSERT INTO solicitacao (encomenda_fk, produto_fk, fornecedor_fk, data_encomenda, data_chegada, quantidade_solicitacao, custo_solicitacao) VALUES
(4, 1, 1, '2019-12-22 00:00:00', '2019-12-26 00:00:00', 50, 50),
(1, 4, 1, '2019-12-22 00:00:00', '2019-12-26 00:00:00', 30, 50),
(1, 7, 1, '2019-12-22 00:00:00', '2019-12-26 00:00:00', 40, 120),
(1, 8, 1, '2019-12-22 00:00:00', '2019-12-26 00:00:00', 30, 40),
(2, 4, 1, '2019-12-22 00:00:00', '2019-12-27 12:18:27', 500, 30),
(2, 5, 4, '2019-12-22 00:00:00', '2019-12-27 12:18:27', 500, 100),
(2, 10, 4, '2019-12-22 00:00:00', '2019-12-27 12:18:27', 1000, 5),
(3, 7, 6, '2019-12-29 10:01:17', '2019-12-31 17:32:04', 30, 35),
(3, 12, 6, '2019-12-29 10:01:17', '2019-12-31 17:32:04', 250, 2.7),
(4, 1, 3, '2019-12-31 15:26:18', '2019-12-31 17:29:43', 4500, 35),
(4, 14, 3, '2019-12-31 15:26:18', '2019-12-31 17:29:43', 4500, 35),
(5, 2, 5, '2019-12-31 23:13:57', '2019-12-31 23:17:13', 407, 7),
(5, 4, 5, '2019-12-31 23:13:57', '2019-12-31 23:17:13', 40, 34);
```

## ● Contextualização:

Os comandos LMD utilizados no SQL, nesse caso, o comando INSERT.

O comando INSERT permite realizar a introdução de novos registos numa tabela da base de dados.

## ● Queries:

# CONCLUSÕES

Com o passar do tempo esta base de dados teve diversas versões, de início parecia haver uma solução simples mas logo foi percebido que existiam muitas e na base de tentativa e erro chegamos ao melhor entendimento de como funcionaria uma base de dados com os objetivos apresentados. Conforme chegavamos nesse entendimento a base de dados começava a fazer mais sentido e se “encaixava” nos requisitos necessários para a implementação da mesma no projeto conjunto.

Neste documento foram abordadas todas as dependências e associações em cada um dos modelos da base de dados, explicando a fundo as decisões acerca dela.