

INSTITUTO DE EDUCAÇÃO SUPERIOR DA PARAÍBA BACHARELADO  
EM SISTEMAS DE INFORMAÇÃO

Angelo Carvalho

Eduardo Galdino

Sinésio Vinicius

Proposta de um Modelo de Banco de Dados Comercial Logístico na  
Área de Games

CABEDELO

2019

Angelo Carvalho

Eduardo Galdino

Sinésio Vinicius

Proposta de um Modelo de Banco de Dados Comercial Logístico na  
Área de Games

Trabalho apresentado ao Instituto de Educação Superior da Paraíba, no curso Bacharelado em Sistemas de Informação, para obtenção de uma nota na disciplina de Banco de Dados I sob orientação do Professor Fábio Nicácio de Medeiros.

CABEDELO  
2019

## SUMÁRIO

1	INTRODUÇÃO .....	4
1.1	Objetivos.....	4
1.2	Metodologia do Trabalho .....	4
2	REGRAS DE NEGÓCIO.....	4
3	MODELO CONCEITUAL .....	5
4	Modelo Lógico .....	8
5	Dicionário de Dados.....	11
6	Restrições .....	15
7	Script de criação .....	18
8	Consultas SQL.....	21
9	Stored Procedures.....	23
10	Considerações finais .....	24
11	Referências Bibliográficas.....	25

## 1 INTRODUÇÃO

O comércio é um elemento fundamental e importante dentro da sociedade. Um grande impulsionador para o aumento de economia em qualquer lugar do mundo. Nesse sentido, com a evolução da tecnologia a categoria o mercado de games vem se fortalecendo a cada ano. Esse projeto tem o intuito de viabilizar e facilitar a relação do comércio com o cliente, assim como desenvolver um sistema que seja capaz de atender as necessidades do mesmo.

A empresa terá um conteúdo voltado para a área de games possuindo um sistema que automatize e viabilize sua relação com o cliente. Nesse sentido, o cliente (Pessoa física ou Jurídica) escolhe o produto tem opção de fazer essa escolha fisicamente (Na loja em determinado local) e virtualmente (Fazendo cadastro e login no sistema da empresa). Para suprir as necessidades da empresa com a demanda de produtos, há certos fornecedores de gêneros diferentes levando categorias diversas de produtos a empresa.

### 1.1 Objetivos

Esse projeto tem como objetivo distinguir uma base teórica e conceitual do negócio, dentre os quais os destaques das etapas essenciais para compor o projeto negócio de pesquisa para o trabalho do banco de dados referente ao comércio relacionado a área de Games. Será possível ter a base de como o sistema funcionará, contendo todas as informações necessárias para que o projeto se inicie.

### 1.2 Metodologia do Trabalho

O estudo foi desenvolvido com base ao conceito regras de negócio dentro de uma logística do comércio relacionado a Games. Utilizando-se pesquisas relevantes através de conceitos fundamentais do Banco de Dados, conceitos e modos de tocar o negócio. Assim como também foram utilizadas definições e ferramentas utilizadas em sala de aula para a conclusão desse projeto.

## 2 REGRAS DE NEGÓCIO

O cliente pode ter a opção de escolher entre comprar na loja física ou virtual. Nesse sentido, na física irá ter um funcionário onde ele irá perguntar se você é cadastrado no site da empresa e se já tem o login com todos os seus dados, pois viabiliza o processo. Logo após, o cliente tendo em mente em comprar algum produto, irá escolher o mesmo e o funcionário tem a função de atendê-lo e informá-lo com as melhores informações sobre o produto que o cliente escolheu. Já na loja virtual, o cliente poderá se cadastrar facilmente no site, utilizando senha e login, estará com acesso a toda plataforma online de produtos da empresa. Ao decorrer do processo se o cliente se interessar por algum produto, terá como opção pagar via boleto e cartão de crédito, vinculado com sua compra, iremos precisar do endereço do mesmo para assim fazer a entrega do produto. Nesse sentido, o cliente poderá optar por Sedex ou PAC, dependendo de sua situação. Desse modo, o projeto foi implementado para viabilizar e facilitar o relacionamento entre empresa e consumidor.

### 3 MODELO CONCEITUAL

É uma representação fiel da realidade observada dentro de um modelo de relacionamento entre entidades. Desse modo, é composto por relacionamentos, atributos e cardinalidades dentre as quais são cruciais para definições de características desse relacionamento. Nesse modelo é possível identificar, juntamente com o cliente, quais as entidades, relacionamentos, atributos e cardinalidades que serão utilizadas para o funcionamento do produto de acordo com a necessidade do cliente.

Torna-se importante que abstração de todas as informações seja a mais exata possível para que não ocorra a necessidade de futura correções, o que resultará em mais tempo de desenvolvimento e um aumento no custo. É no modelo conceitual que fica registrado os dados que serão encontrados no banco de dados. A prática mais conhecida é a abordagem entidade relacionamento (ER) que utiliza de um diagrama para representa-lo, conhecido como diagrama entidade-relacionamento. Posteriormente será possível visualizá-lo.

#### Entidades

Formam um conjunto de “coisas” com conceitos comuns às quais desejamos armazenar os dados. Na ferramenta utilizada, a entidade é representada por um retângulo, podendo ter seus atributos atrelados a ela, que irão identificar e descrever essa entidade, permitindo que cada entidade seja única e “armazene” apenas os dados necessários.



Figura 1 - Demonstração de entidades

Essas entidades são classificadas entre entidades fortes e fracas. A que possui uma chave primária é considerada entidade forte, se não for possível gerar uma chave primária, será uma entidade fraca (depende de outra), que é representada por 2 retângulos.

#### Relacionamentos

São associações entre entidades com um significado específico dentro do mundo real. Essas informam o que irá acontecer entre as entidades que estão se relacionando. No exemplo é demonstrado que o cliente realiza nenhum ou vários pedidos, o que mostra a importância de ser demonstrado como esse relacionamento funcionará por meio da cardinalidade. É importante que seja utilizado apenas uma palavra, sendo preferencialmente um verbo, para que haja uma rápida compreensão do que está acontecendo.



Figura 2 - Exemplo de um relacionamento entre entidades

### Atributos

São as características individuais que descrevem cada entidade. Esses atributos serão os responsáveis por caracterizar a entidade, permitindo que os atributos representem a entidade, também permitem que a informação correspondente seja organizada de acordo com o atributo certo, no campo certo, para que haja a formação de tabelas, essas que permitirão os diversos relacionamentos.

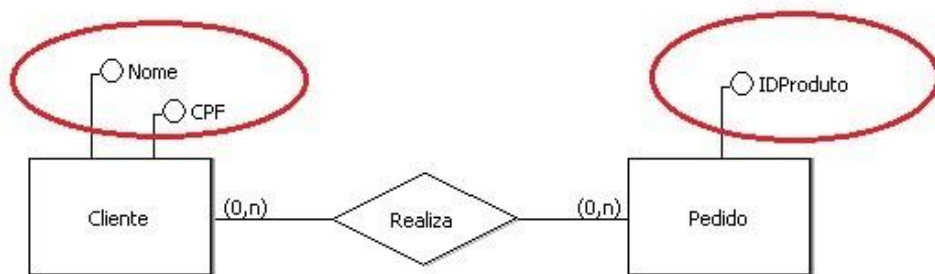


Figura 3 - Atributos das entidades

### Cardinalidades

São as quantidades de informações que podem percorrer entre as entidades. Permitindo uma interação entre as entidades, fazendo com que esses atributos e tabelas possam ter uma relação. Também é possível informar como os dados serão armazenados. Deve-se levar em consideração os exemplos criados no modelo conceitual para o desenvolvimento do modelo lógico.

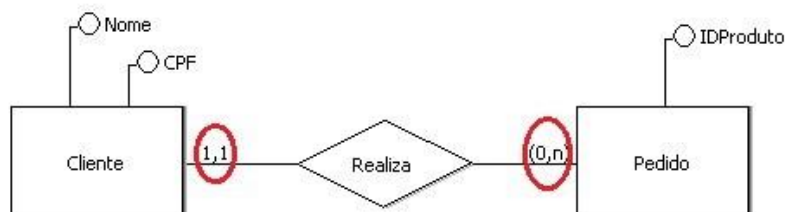


Figura 4 - Exemplo de cardinalidade

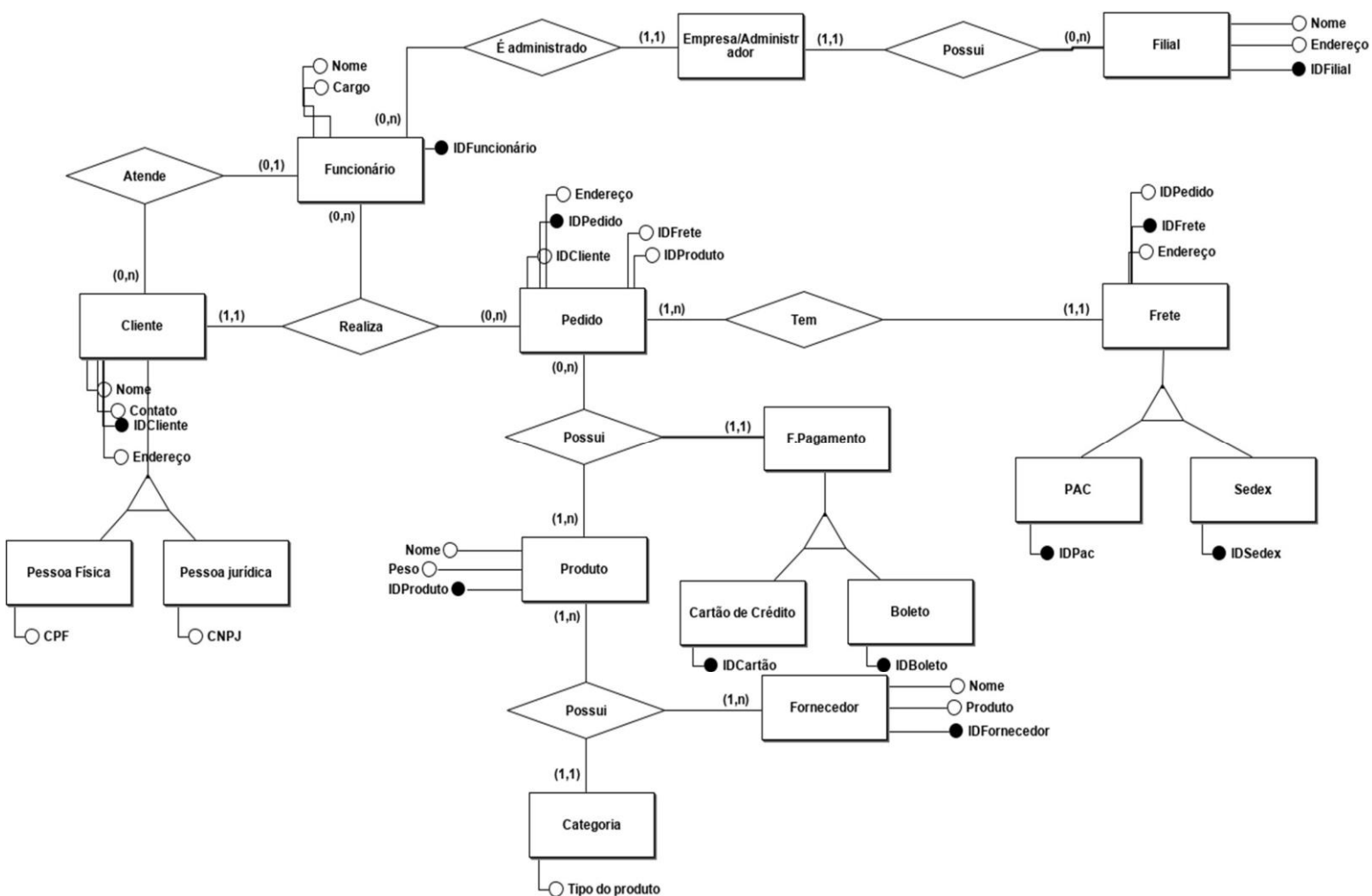


Figura 5 - Modelo conceitual

Foram utilizadas diversas entidades para a construção do modelo conceitual. As entidades **CLIENTE**, **FORMAS DE PAGAMENTO** e **FRETE** possuem uma especialização do tipo total, indicando que as entidades que estarão envolvidas terão que ser utilizadas, exemplo, um cliente deve ser do tipo pessoa física ou pessoa jurídica, assim como a forma de pagamento deverá ser por meio de cartão de crédito ou boleto, a mesma coisa acontecerá com o frete, será pac ou sedex.

## 4 Modelo Lógico

O modelo lógico toma como base o modelo conceitual, que fornecerá as informações necessárias para a sua construção. Nesse modelo serão informadas principalmente as chaves primárias, candidatas e estrangeiras, o que permitirá uma maior interação entre as entidades, permitindo que esses atributos e tabelas possam ter uma relação. Também é possível informar como os dados serão armazenados. Deve-se levar em consideração os exemplos criados no modelo conceitual para o desenvolvimento do modelo lógico.

Quando se trata de um banco de dados relacional, o modelo lógico é quem define as tabelas que serão utilizadas. Leva-se em consideração que o modelo lógico é uma descrição do banco de dados vista pelo usuário do SGBD, que permitirá que esse modelo seja usado no SGBD escolhido. Esse usuário irá otimizar o modelo lógico para funcionar da melhor maneira possível com o sistema que ele será utilizado.

Além disso, existem os tipos de chaves que irão definir conceitos dentro de um banco de dados, conceitos estes de identificação nas tabelas formadas, contendo regras exclusivas de base de dados. Desse modo, Uma chave em Bases de dados é uma coluna da tabela que permite, para cada linha, identificar essa linha (caso seja a chave primária), ou relacionar com outra tabela no caso da chave estrangeira.

### Chave Primária

É um atributo que servirá para identificar as informações contidas em uma tabela. Uma chave primária é uma regra implementada em uma coluna ou em um conjunto de colunas de forma a garantir que os valores contidos nesta (s) sejam únicos, ou seja, estes valores nunca irão se repetir.

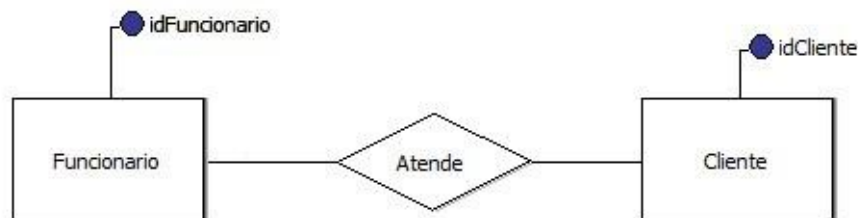


Figura 6 - Exemplo de PK

### Chave Candidata

Essa chave é formada por um atributo ou mais, permitindo a identificação de cada linha de uma tabela. Pode ser utilizada para marcar um atributo único



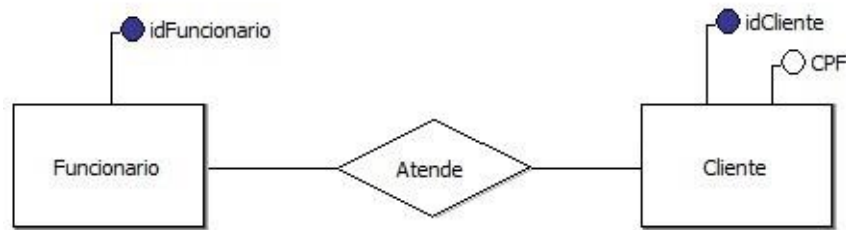


Figura 7 - Exemplo de Chave candidata.

Na figura acima, o que se destaca na chave alternativa é como podemos utiliza-la. No campo CPF na tabela cliente, utilizando a chave candidata, os números de CPF do cliente, não poderão ser os mesmos, isso explica basicamente o conceito de uma chave candidata.

### Chave Estrangeira

A chave estrangeira permite o relacionamento entre tabelas, ela corresponderá ao atributo referencial de uma entidade. Ela pode ser usada como referência por outra tabela, permitindo que a chave primária de outra tabela seja utilizada.

### Restrição de integridade

Série de regras que poderão fazer com que as informações sejam armazenadas de forma correta e organizadas.

### Integridade Semântica

Permite que a regra de negócio seja seguida a risca, fazendo com que o modelo lógico trabalhe como foi planejado na regra de negócio.

### Integridade Referencial

Possui uma série de regras que permitem que os dados que forem referenciados devem estar corretos, sem sofrer alterações. Uma chave estrangeira deverá ser a chave primária da tabela de origem.

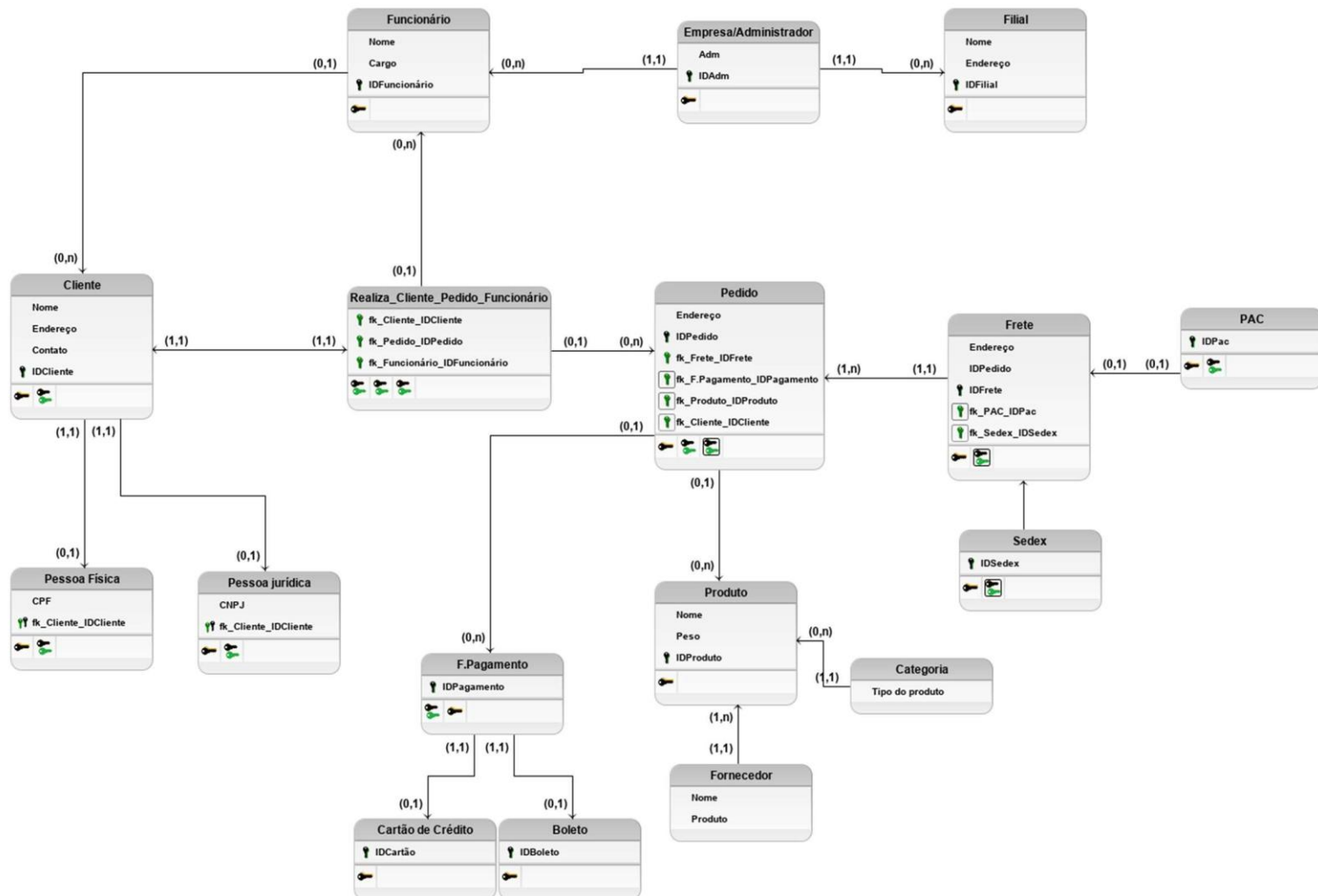


Figura 8 - Modelo lógico

A tabela Pedido terá o seu ID, mas também receberá informações de outras tabelas como produto, f. pagamento, frete, cliente. Todas essas informações que serão obtidas por meio da utilização de chaves estrangeiras poderão ser usadas para organizar todas as informações necessárias para se gerar um pedido, fazendo com que tudo que esteja relacionado ao pedido seja usado. Também será gerada uma tabela com o relacionamento entre funcionário, cliente e pedido, permitindo que seja possível a identificação se foi um cliente ou um funcionário que realizou certo pedido.

## 5 Dicionário de Dados

O Dicionário de dados consiste em reunir todos os elementos de dados que vão ser utilizados. Nesse sentido, sem o dicionário de dados o modelo do banco de dados não pode ser considerado completo, pois o mesmo descreve as entradas, saídas e composição de depósitos. Desse modo, cada entrada do dicionário de dados é constituída por um atributo identificador e seu respectivo tipo de descrição. A descrição do dicionário de dados inclui o significado, conteúdo, valores permitidos, unidades e chave primária. O Dicionário de dados consiste em fazer referência a todos os elementos envolvidos na medida em que permite associar um significado a cada termo utilizado.

TABELA CLIENTE

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDCliente	Char(5)	não	Código do cliente		X		
Nome	Varchar(40)	não	Nome do cliente				
CPF	Varchar(12)	Não	CPF DO CLIENTE				X
Endereco	Varchar(50)	não	Endereço(rua,número,bairro)				
Fone	Varchar(16)	não	Número de telefone				

TABELA FUNCIONÁRIO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDFuncionario	Char(5)	não	Código do funcionário		X		
Nome	Varchar(40)	não	Nome do funcionário				
Cargo	Varchar(30)	não	Cargo do funcionário				
Sexo	Char(1)	não	Sexo do funcionário	F ou M			
RG	Varchar(15)	não	RG do funcionário				X
CPF	Varchar(12)	não	CPF do funcionário				
Salario	smallmoney	sim	Salário do funcionário				

TABELA FILIAL

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDFilial	Char(5)	não	Código da filial		X		
Nome	Varchar(40)	não	Nome da filial				
Endereco	Varchar(50)	não	Endereço da filial				
CNPJ	Varchar(14)	não	CNPJ da filial				X

TABELA PEDIDO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDPedido	char(5)	não	Código do pedido		X		
Endereco	Varchar(50)	não	Endereço do pedido				
IDCliente	Char(5)	não	Código do cliente	Tabela Cliente		X	
IDFrete	Char(5)	não	Código do frete	Tabela Frete		X	
IDProduto	Char(5)	não	Código do produto	Tabela Produto		X	

TABELA FRETE

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDFrete	Char(5)	não	Código do frete		X		
IDPedido	Char(5)	não	Código do pedido	Tabela Pedido		X	
Endereco	Varchar(50)	não	Endereço do frete				

TABELA PRODUTO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDProduto	Char(5)	não	Código do produto		X		
Nome	Varchar(30)	não	Nome do Produto				X
Peso	Smallint	não	Peso do produto				
Descricao	Varchar(50)	não	Descrição do produto				
Custo	Smallmoney	não	Valor da compra				
venda	Smallmoney	não	Valor da venda				
tipo	Char(5)	não	Categoria do produto			X	
quant	Smallint	não	Quantidade no estoque				

TABELA PAC

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDPac	Char(5)	não	Código do pac		X		

TABELA SEDEX

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDSedex	Char(5)	não	Código do sedex		X		

TABELA PESSOA FÍSICA

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
cpf	Char(12)	não	Cpf do cliente			X	

TABELA PESSOA JURÍDICA

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
Cnpj	Char(12)	não	Cnpj da empresa			X	

TABELA FORMA DE PAGAMENTO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDPagamento	Char(5)	não	Código forma de pagamento		X		

TABELA CARTÃO DE CRÉDITO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMINIO	CHAVE		
					PRI	EST	CAN
IDcartao	Char(5)	não	Código do cartão de credito		X		

TABELA BOLETO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					PRI	EST	CAN
IDboleto	Char(5)	não	Código boleto		X		

TABELA FORNECEDOR

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					PRI	EST	CAN
IDFornecedor	Char(5)	Não	Código do fornecedor		X		
Nome	Varchar(30)	Não	Nome do fornecedor				
Produto	Varchar(30)	Não	Produto fornecido				

TABELA TIPO

ATRIBUTO	TIPO	NULO	DESCRIÇÃO	DOMÍNIO	CHAVE		
					PRI	EST	CAN
IDTipo	Char(5)	não	Código do tipo		X		
Nome	Varchar(20)	não	Nome do tipo				X
Descrição	Text	sim	Descrição do tipo				

Entidade: é o nome da entidade que foi definida no MER.

Atributos: São os atributos da entidade.

Classes: As classes podem ser: simples, composto, multivalorado e determinante.

Domínio: podem ser numéricos, texto, data e booleano.

Tamanho: Define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo. Geralmente o tamanho é definido apenas para atributos de domínio texto. Descrição: é opcional e pode ser usado para descrever o que é aquele atributo.

Nos campos Domínio e Descrição lembrar sempre de usar a notação definida pelo Yourdon. É através desta notação que será possível definir com precisão (formalizar) o formato do dado a ser armazenado pelo campo.

Pode-se utilizar o Excel (ou qualquer outro processador de planilhas) ou o próprio Word para elaborar as tabelas que devem descrever todas as relações do seu modelo relacional como mostrado nos exemplos acima.

## 6 Restrições

As restrições servem para deixar claro o que e como um campo será acrescentado no banco de dados, garantindo que tudo esteja bem definido, seja o que será chave primária, secundária, estrangeira e etc. É de extrema importância que um banco de dados tenha dados corretos, que representem e funcionem de acordo com o desejado, por isso a identificação de todas as restrições presentes no projeto se torna um dos papéis mais importantes, que permitirá com que esses dados sejam verídicos. Se for necessário identificar que tipo de caractere um campo receberá, é possível por meio das restrições, para que a equipe entenda que certo campo receberá 5 letras, números, ou que será um campo de referência.

### Cliente

Coluna	Tipo	Nome	Expressão
IDCLIENTE	Chave primária	PK_CLIENTE	Primary Key
IDCLIENTE	Permitido: 5 letras	CK_CLIENTE_IDCLIENTE	CHECK ( código LIKE '[A-Z][A-Z] [A-Z][A-Z][A-Z]')
CPF	Chave candidata	UN_CLIENTE_CPF	UNIQUE

### Funcionario

Coluna	Tipo	Nome	Expressão
IDFUNCIONARIO	Chave Primária	PK_FUNCIONARIO	Primary Key
SEXO	Permitidos: F ou M	CK_FUNCIONARIO_SEXO	CHECK ( sexo LIKE '[FM]' )
RG	Chave Candidata	UN_FUNCIONARIO_RG	UNIQUE

### FILIAL

Coluna	Tipo	Nome	Expressão
IDFILIAL	Chave primária	PK_FILIAL	Primary Key
CNPJ	Chave candidata	UN_FILIAL_CNPJ	UNIQUE

### Pedido

Coluna	Tipo	Nome	Expressão
IDPEDIDO	Chave Primária	PK_PEDIDO	Primary Key
IDCLIENTE	Chave Estrangeira para referenciar a tabela cliente	FK_PEDIDO_IDCLIENTE	FOREIGN KEY REFERENCES CLIENTE(IDCLIENTE)

#### FRETE

Coluna	Tipo	Nome	Expressão
IDFRETE	Chave Primária	PK_FRETE	Primary Key
IDPEDIDO	Chave Estrangeira	FK_FRETE_IDPEDIDO	FOREIGN KEY REFERENCES PEDIDO(IDPEDIDO)

#### TIPO

Coluna	Tipo	Nome	Expressão
IDTIPO	Chave Primária	PK_TIPO	Primary Key
NOME	Chave candidata	UM_TIPO_NOME	UNIQUE

#### Produto

Coluna	Tipo	Nome	Expressão
IDPRODUTO	Chave Primária	PK_PRODUTO	Primary Key
TIPO	Chave Estrangeira	FK_PRODUTO	FOREIGN KEY REFERENCES TIPO(IDTIPO)

#### PAC

Coluna	Tipo	Nome	Expressão
IDPAC	Chave Primária	PK_PAC	Primary Key

#### Sedex

Coluna	Tipo	Nome	Expressão
IDSEDEX	Chave Primária	PK_Sedex	Primary Key

#### PessoaFisica

Coluna	Tipo	Nome	Expressão
IDPESSOAFISICA	Chave Primária	PK_PESSOAFISICA	Primary Key

#### PessoaJuridica

Coluna	Tipo	Nome	Expressão
IDPESSOAJURIDICA	Chave Primária	PK_PESSOAJURIDICA	Primary Key

#### FormadePagamento

Coluna	Tipo	Nome	Expressão
IDPAGAMENTO	Chave Primária	PK_IDPAGAMENTO	Primary Key

#### CartaoDeCredito

Coluna	Tipo	Nome	Expressão
IDCARTAO	Chave Primária	PK_CARTAODECREDITO	Primary Key



Boleto

Coluna	Tipo	Nome	Expressão
IDBoleto	Chave Primária	PK_BOLETO	Primary Key

Fornecedor

Coluna	Tipo	Nome	Expressão
IDFORNECEDOR	Chave Primária	PK_FORNECEDOR	Primary Key

## 7 Script de criação

*Script* é um conjunto de instruções em código, ou seja, escritas em linguagem de computador. É uma linguagem de programação que executa diversas funções no interior de um programa de computador. Nele, estão contidas todas as informações sobre o projeto ou o que você está desenvolvendo. A importância do mesmo, serve para aplicação do que você está planejando para traçar e concluir seu projeto. A seguir, está o script do ProjetoLoja:

Estamos criando na base de dados o nosso projeto, que se denomina ProjetoLoja, com um tamanho de 5 até 10, com um crescimento de arquivo de 1;

```
CREATE DATABASE PROJETOLOJA
ON(NAME = PROJETOLOJA,
  FILENAME = 'C:\BANCO_DADOS\PROJETOLOJA.MDF',
  SIZE = 5,
  MAXSIZE = 10,
  FILEGROWTH = 1
)

USE PROJETOLOJA
```

Após a criar e selecionar o banco de dados, podemos começar a criar as tabelas e suas relações.

```
CREATE TABLE CLIENTE (
  IDCLIENTE CHAR(5) NOT NULL
    CONSTRAINT PK_CLIENTE PRIMARY KEY,
  NOME VARCHAR(40) NOT NULL, --check do nome
  CPF VARCHAR(12) NOT NULL
    CONSTRAINT UN_CLIENTE_CPF UNIQUE, -- check 12 digitos
  ENDereco VARCHAR(50) NOT NULL,
  FONE VARCHAR(16) NOT NULL
)

CREATE TABLE FUNCIONARIO(
  IDFUNCIONARIO CHAR(5) NOT NULL
    CONSTRAINT PK_FUNCIONARIO PRIMARY KEY,
  NOME VARCHAR(40) NOT NULL, --check nome
  CARGO VARCHAR(30) NOT NULL,
  SEXO CHAR(1) NOT NULL
    CONSTRAINT CH_FUNCIONARIO_SEXO CHECK (SEXO = 'F' OR SEXO = 'M'),
  RG VARCHAR(9) NOT NULL
    CONSTRAINT UN_FUNCIONARIO_RG UNIQUE,
  CPF VARCHAR(12) NOT NULL --check 12 digitos
)

Alter Table Funcionario
ADD SALARIO SMALLMONEY
```

Adicionamos um campo na tabela Funcionário para poder receber o salário do profissional.

```

CREATE TABLE FILIAL(
    IDFILIAL CHAR(5) NOT NULL
        CONSTRAINT PK_FILIAL PRIMARY KEY,
    NOME VARCHAR(40) NOT NULL,
    ENDereco VARCHAR(50) NOT NULL,
    CNPJ VARCHAR(14) NOT NULL
        CONSTRAINT UN_FILIAL_CNPJ UNIQUE
)

CREATE TABLE PEDIDO(
    IDPEDIDO CHAR(5) NOT NULL
        CONSTRAINT PK_PEDIDO PRIMARY KEY,
    ENDereco VARCHAR(50) NOT NULL,
    IDCLIENTE CHAR(5) NOT NULL
        CONSTRAINT FK_PEDIDO_CLIENTE FOREIGN KEY
        REFERENCES CLIENTE(IDCLIENTE),
    FRETE SMALLMONEY NOT NULL
)

ALTER TABLE PEDIDO ADD IDFUNCIONARIO CHAR(5)
        CONSTRAINT FK_PEDIDO_FUNCIONARIO FOREIGN KEY
        REFERENCES FUNCIONARIO(IDFUNCIONARIO)

ALTER TABLE PEDIDO
ADD DATAPEDIDO SMALLDATETIME

```

Foi criado um campo IDFUNCIONARIO para poder ser usado para referenciar um funcionário ao pedido, assim como também criamos um campo para armazenar a data do pedido.

```

CREATE TABLE TIPO(
    IDTIPO CHAR(5) NOT NULL
        CONSTRAINT PK_TIPO PRIMARY KEY,
    NOME VARCHAR(20) NOT NULL
        CONSTRAINT UN_PAIS_NOME UNIQUE,
    DESCRICAO TEXT NULL
)

CREATE TABLE PRODUTO (
    IDPRODUTO CHAR(5) NOT NULL
        CONSTRAINT PK_PRODUTO PRIMARY KEY,
    NOME VARCHAR(30) NOT NULL,
    PESO SMALLINT NOT NULL,
    DESCRICAO VARCHAR(50) NOT NULL,
    CUSTO SMALLMONEY NOT NULL,
    VENDA SMALLMONEY NOT NULL,
    TIPO CHAR(5) NOT NULL
        CONSTRAINT FK_PRODUTO_TIPO FOREIGN KEY
        REFERENCES TIPO(IDTIPO)
)

ALTER TABLE PRODUTO ADD IDFORNECEDOR CHAR(5)
        CONSTRAINT FK_PRODUTO_FORNECEDOR FOREIGN KEY
        REFERENCES FORNECEDOR(IDFORNECEDOR)

```

Campo adicionado para identificar o fornecedor de cada produto.

```
ALTER TABLE PEDIDO ADD IDPRODUTO CHAR(5) NOT NULL
                        CONSTRAINT FK_PEDIDO_PRODUTO FOREIGN KEY
                        REFERENCES PRODUTO(IDPRODUTO)
```

IDPRODUTO criado na tabela pedido para referenciar o produto, o que permitirá visualizar o produto no pedido.

```
CREATE TABLE PAC(
    IDPAC CHAR(5) NOT NULL
        CONSTRAINT PK_PAC PRIMARY KEY
)
```

```
CREATE TABLE SEDEX(
    IDSEDEX CHAR(5) NOT NULL
        CONSTRAINT PK_SEDEX PRIMARY KEY
)
```

```
CREATE TABLE PESSOAFISICA(
    IDPESSOAFISICA CHAR(5) NOT NULL
        CONSTRAINT PK_PESSOAFISICA PRIMARY KEY
)
```

```
CREATE TABLE PESSOAJURIDICA(
    IDPESSOAJURIDICA CHAR(5) NOT NULL
        CONSTRAINT PK_PESSOAJURIDICA PRIMARY KEY
)
```

```
CREATE TABLE FORMADEPAGAMENTO(
    IDPAGAMENTO CHAR(5) NOT NULL
)
```

```
CREATE TABLE CARTAODECREDITO(
    IDCARTAO CHAR(5) NOT NULL
)
```

```
CREATE TABLE BOLETO(
    IDBOLETO CHAR(5) NOT NULL
)
```

```
CREATE TABLE FORNECEDOR(
    IDFORNECEDOR CHAR(5) NOT NULL
        CONSTRAINT PK_FORNECEDOR PRIMARY KEY,
    NOME VARCHAR(30) NOT NULL,
    PRODUTO VARCHAR(30) NOT NULL
)
```

Por fim, concluímos nosso Script, criando todas as tabelas, atribuindo seus requisitos e definindo suas restrições.

## 8 Consultas SQL

Abaixo estão as nossas consultas para poder visualizar o que foi inserido no nosso banco de dados

```
-- consultar o id do produto, seu nome e seu fornecedor
SELECT IDPRODUTO,NOME,IDFORNECEDOR FROM PRODUTO

-- consultar o pedido e o cliente por meio de um inner join
SELECT PEDIDO.IDPEDIDO, CLIENTE.NOME
FROM PEDIDO
INNER JOIN CLIENTE
ON CLIENTE.IDCLIENTE = PEDIDO.IDCLIENTE
ORDER BY PEDIDO.IDPEDIDO

--consultar o número do pedido, cliente e o nome do produto utilizando inner join para
acessar ambas as tabelas ao mesmo tempo
SELECT PEDIDO.IDPEDIDO, CLIENTE.NOME, PRODUTO.NOME, PEDIDO.ENDERECO 'Nome do produto'
FROM PEDIDO
INNER JOIN CLIENTE
ON CLIENTE.IDCLIENTE = PEDIDO.IDCLIENTE
INNER JOIN PRODUTO
ON PRODUTO.IDPRODUTO = PEDIDO.IDPRODUTO
ORDER BY PEDIDO.IDPEDIDO

-- consultar o produto e seu fornecedor
SELECT PRODUTO.IDPRODUTO, PRODUTO.NOME 'Nome do Produto', PRODUTO.IDFORNECEDOR,
FORNECEDOR.NOME
FROM PRODUTO
INNER JOIN FORNECEDOR
ON FORNECEDOR.IDFORNECEDOR = PRODUTO.IDFORNECEDOR

-- consultar produtos e seus preços
SELECT NOME, CUSTO, VENDA FROM PRODUTO

-- Exibir o preço total de venda dos produtos
SELECT SUM(VENDA) 'VALOR TOTAL DE VENDA' FROM PRODUTO

-- Exibir o preço do teclado(codigo DDDDD) mais barato
SELECT MIN(CUSTO) 'Custo do teclado' FROM PRODUTO WHERE TIPO = 'DDDDD'

-- Exibir salários dos funcionários
SELECT IDFUNCIONARIO, NOME, CARGO, SALARIO FROM FUNCIONARIO

-- Exibir Pedido com o Funcionário que realizou o pedido
SELECT PEDIDO.IDPEDIDO, PEDIDO.ENDERECO, CLIENTE.NOME, PEDIDO.FRETE, PEDIDO.IDPRODUTO,
PEDIDO.DATAPEDIDO, FUNCIONARIO.NOME
FROM PEDIDO
INNER JOIN FUNCIONARIO
ON PEDIDO.IDFUNCIONARIO = FUNCIONARIO.IDFUNCIONARIO
INNER JOIN CLIENTE
ON CLIENTE.IDCLIENTE = PEDIDO.IDCLIENTE

--Exibir produtos, tipos e fornecedores
SELECT PRODUTO.NOME 'Nome', TIPO.NOME 'Tipo', FORNECEDOR.NOME
FROM PRODUTO
JOIN TIPO
ON TIPO.IDTIPO = PRODUTO.TIPO
JOIN FORNECEDOR
ON FORNECEDOR.IDFORNECEDOR = PRODUTO.IDFORNECEDOR
```

```
--Exibir Produtos e o tipo  
SELECT PRODUTO.NOME 'Nome', TIPO.NOME 'Tipo'  
FROM PRODUTO  
JOIN TIPO  
ON TIPO.IDTIPO = PRODUTO.TIPO
```

## 9 Stored Procedures

Stored procedures são utilizadas para “automatizar” certos códigos, possibilitando o reuso de consultas, fazendo com que o usuário não precise toda vez procurar por uma consulta ou que seja necessário digitar tudo novamente, apenas com um simples EXEC é possível executar o script criado.

```
--Exibir Produtos e o tipo
CREATE PROCEDURE spExibirProdutosTipos
AS SELECT PRODUTO.NOME 'Nome', TIPO.NOME 'Tipo'
FROM PRODUTO
JOIN TIPO
ON TIPO.IDTIPO = PRODUTO.TIPO

EXEC spExibirProdutosTipos

-- Exibir salários dos funcionários
CREATE PROCEDURE spExibirSalarios
AS SELECT Nome, SALARIO 'Salário'
FROM FUNCIONARIO

EXEC spExibirSalarios

--Exibir produtos tipos e fornecedores
CREATE PROCEDURE spExibirProdutosTiposFornecedor
AS SELECT PRODUTO.NOME 'Nome', TIPO.NOME 'Tipo', FORNECEDOR.NOME
FROM PRODUTO
JOIN TIPO
ON TIPO.IDTIPO = PRODUTO.TIPO
JOIN FORNECEDOR
ON FORNECEDOR.IDFORNECEDOR = PRODUTO.IDFORNECEDOR

EXEC spExibirProdutosTiposFornecedor

-- exibir número do pedido, nome do cliente e o nome do funcionário
CREATE PROCEDURE spExibirPedidoClienteFuncionario
AS SELECT PEDIDO.IDPEDIDO, CLIENTE.NOME, FUNCIONARIO.NOME 'Funcionário',
PEDIDO.DATAPEDIDO
FROM PEDIDO
JOIN CLIENTE
ON CLIENTE.IDCLIENTE = PEDIDO.IDCLIENTE
JOIN FUNCIONARIO
ON FUNCIONARIO.IDFUNCIONARIO = PEDIDO.IDFUNCIONARIO

EXEC spExibirPedidoClienteFuncionario

--Exibir funcionário e cargo depois de inserir o cargo
CREATE PROCEDURE spExibirFuncionario
@CARGO VARCHAR(30)
AS SELECT FUNCIONARIO.NOME 'NOME', FUNCIONARIO.CARGO 'CARGO'
FROM FUNCIONARIO WHERE CARGO = @CARGO

EXEC spExibirFuncionario 'ATENDENTE'
EXEC spExibirFuncionario 'ADMINISTRADOR'
```

## 10 Considerações finais

Este projeto se propôs, com o objetivo geral de praticar os conhecimentos em banco de dados e aplicar o que foi estudado dentro e fora de sala de aula, assim como transmitir por meio deste trabalho a diversidade de características e definições de um banco de dados e seus modelos que o compõe. Também foi possível ter uma base melhor de banco de dados, permitindo a emissão desta documentação com uma demonstração dos diferentes tipos de modelos, assim como imagens que demonstram o modelo conceitual e lógico. Foi possível montar uma base que poderá ser utilizada para a criação do sistema proposto, seguindo conforme o plano de negócio definido.



## 11 Referências Bibliográficas

ALVES, William Pereira. Banco de dados, 1ª Edição. 2013.

HEUSER, A. C. Projeto de Banco de Dados, 6ª edição. ARTMED EDITORA S.A. 2009.

[https://www.ibm.com/support/knowledgecenter/en/SS62YD\\_4.1.1/com.ibm.datatools.logical.ui.doc/topics/clogmod.html](https://www.ibm.com/support/knowledgecenter/en/SS62YD_4.1.1/com.ibm.datatools.logical.ui.doc/topics/clogmod.html)

<https://www.guru99.com/data-modelling-conceptual-logical.html>