

INSTITUTO FEDERAL DO PARANÁ  
LUIZ GUSTAVO FRANCISCO  
CHRISTIAN GARCIA  
LUCAS APARECIDO DA SILVA FLORENTINO

**PROJETO DE DADOS  
SEC - SISTEMA E-COMMERCE**

PARANAVAÍ - PR  
2019

LUIZ GUSTAVO FRANCISCO  
CHRISTIAN GARCIA  
LUCAS APARECIDO DA SILVA FLORENTINO

**PROJETO DE DADOS  
SEC - SISTEMA E-COMMERCE**

Trabalho da disciplina de  
Fundamentos de Banco de Dados, do  
curso (TADS) Análise e Desenvolvimento de  
Sistemas do Instituto Federal do Paraná -  
IFPR - Campus Paranavaí.  
Professores: Fabiano Utiyama,  
Késsia Rita da Costa Marchi e Mônica  
Tortorelli Winche Achermann

PARANAVAÍ  
2019

## SUMÁRIO

### **1 INTRODUÇÃO 4**

### **2 DESENVOLVIMENTO 4**

2.1 Conceitos de Banco de Dados 4

2.2 Introdução 6

2.3 Desenvolvimento 7

2.4 Dicionário de Dados 10

2.5 Código SQL - Construção de Tabelas 12

2.6 Código SQL - Inserções 15

2.7 Código SQL - Selects 17

2.8 Prototipação 20

### **3 CONCLUSÃO 23**

3.1 análise de custo 23

## **Conceitos de Banco de Dados**

Para Korth, um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”, podemos dizer com isso que, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, pode-se dizer que se tem um banco de dados. O exemplo mais clássico disso é um sistema de RH de uma empresa, ademais, um banco de dados é uma coleção de dados persistentes, usados pelos sistemas de aplicação de uma determinada empresa [Heuser, 2010]. A função dos bancos de dados é o armazenamento dos mesmos, ou seja, um conjunto de registros consistindo em uma ou várias tabelas, que tem como objetivo organizar e guardar dados utilizando SGBD, que facilita a comunicação desses dados no sistema. O SGBD permite a busca, alteração e exclusão dos dados que consiste nas tabelas do banco de dados..

Um banco de dados relacional é um mecanismo de armazenamento que permite a persistência de dados e opcionalmente implementar funcionalidades. Os bancos de dados relacionais são usados para armazenar a informação requerida por aplicações construídas usando tecnologias procedurais, como FORTRAN, tecnologias orientadas a objetos, como Java, e tecnologias baseadas em componentes como Visual Basic.

Um Sistema de Gerenciamento de Banco de Dados Relacional é um software que controla o armazenamento, recuperação, exclusão, segurança e integridade dos dados em um banco de dados. Um banco de dados relacional armazena dados em tabelas. Essas tabelas são organizadas por colunas, e por sua vez cada coluna armazena um tipo de dado. Os dados de uma “instância” simples são armazenados como linhas.

Um Banco de Dados Objeto Relacional é semelhante a um banco de dados relacional, o que diferencia ele é que possui um modelo de banco de dados orientado a objetos, classes e herança são suportados diretamente nos esquemas do banco de dados e na linguagem de consulta. Além disso, ele suporta extensão do modelo de dados com a personalização de tipos de dados e métodos.

Um Sistema de Gerenciamento de Banco de Dados é um software que possui recursos capazes de manipular as informações do banco de dados e interagir com o usuário, esse software incorpora as funções de definição, recuperação e alteração de dados em um banco de dados, Um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados, são projetados para gerir grandes volumes de informações. O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento das informações do banco de dados.

Restrição de Integridade Referencial é Segundo (CAMARGO, 2001) é a restrição que determina que os valores do campo que aparecem em chave estrangeira devem aparecer primeiro em chave primária da tabela referenciada.

**Referências:**

Heuser, Carlos Alberto. *Projeto de Banco de Dados*. 4 edição. UFRGS: Editora Sagra Luzzatto 1998

<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649> <acessado em 27 de março de 2019>

<https://www.devmedia.com.br/bancos-de-dados-relacionais/20401> <acessado em 27 de março de 2019>

CAMARGO, Luiz Carlos. Restrições de Integridade e regras ativas em banco de dados distribuídos. 2001. 175f. Dissertação (Mestrado) – Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2001. Disponível em:

<https://repositorio.ufsc.br/bitstream/handle/123456789/79996/182871.pdf?sequence=1>. Acessado em: 27 março de 2019.

[http://web.unipar.br/~seinpar/2015/\\_include/artigos/Julio\\_Fernandes\\_Rocha.pdf](http://web.unipar.br/~seinpar/2015/_include/artigos/Julio_Fernandes_Rocha.pdf), <acessado em 27 de março de 2019>

<http://www.di.ufpb.br/valeria/BancoDados/Aula1-IntroducaoSlides.pdf> <acessado em 27 de março de 2019>

## 1 INTRODUÇÃO

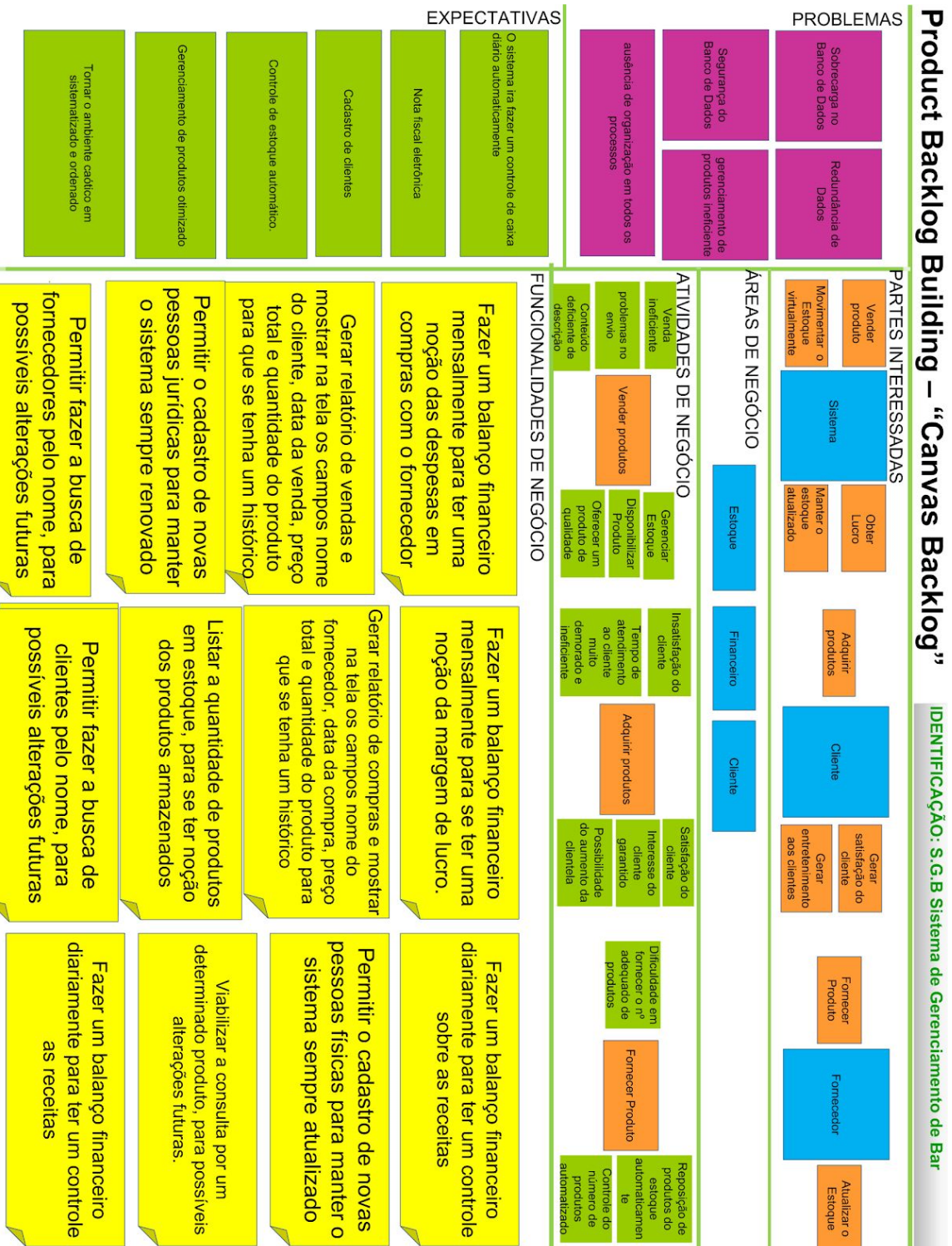
O projeto SEC - Sistema E-commerce surgiu com a proposta de criar um sistema de comércio eletrônico de diversos produtos, lembrando que é apenas uma loja virtual, não existe loja física, o objetivo do sistema é aumentar a organização, eficiência e alcance de clientes, uma loja virtual pode vender muito mais que uma simples loja física, com as técnicas de marketing digital certas, um simples site que permita a venda produtos pode acarretar muito mais tráfego, diferente de lojas físicas, que tem muito mais chance de falirem, de cada dez empresas físicas, seis fecham antes do primeiros 5 anos (IBGE, 2014), além disso, como consequência, o lucro de um E-Commerce, pode ser muito superior, levando em conta que não se tem despesas com muitos funcionários, aluguel e outras contas, essencialmente a cada transação o software atualiza o registro de produtos no Banco de Dados, permitindo o acompanhamento de vendas a clientes e compras de fornecedores em tempo real, a reposição de estoque é controlada pela movimentação dessas transações, com o SEC (Sistema E-Commerce) é muito fácil fazer o controle de receitas, tanto diariamente quanto mensalmente, tornando possível uma certa noção da margem de lucro do site.

## 2 DESENVOLVIMENTO

Referente ao domínio da aplicação se trata de um comércio de produtos diversos completamente digital, O modelo utilizado para desenvolver o projeto foi o Canvas Backlog, manipulado para organizar o levantamento de requisitos, os dados obtidos no Canvas foram coletados a partir de um Brainstorming, essa técnica de extração de requisitos consiste em geração de ideias por uma ou várias reuniões que permite que o integrantes da equipe sugiram ou explorem ideias, quanto mais idéias forem propostas, maior será a chance de aparecerem boas idéias. Os participantes também devem ser encorajados a combinar ou enriquecer as idéias de outros e, para isso, é necessário que todas as idéias permaneçam visíveis a todos os participantes. Na fase final é feita uma revisão das ideias, uma de cada vez. As consideradas valiosas pelo grupo são mantidas e classificadas em ordem de prioridade.

A seguir serão apresentados os requisitos funcionais no Canvas backlog:

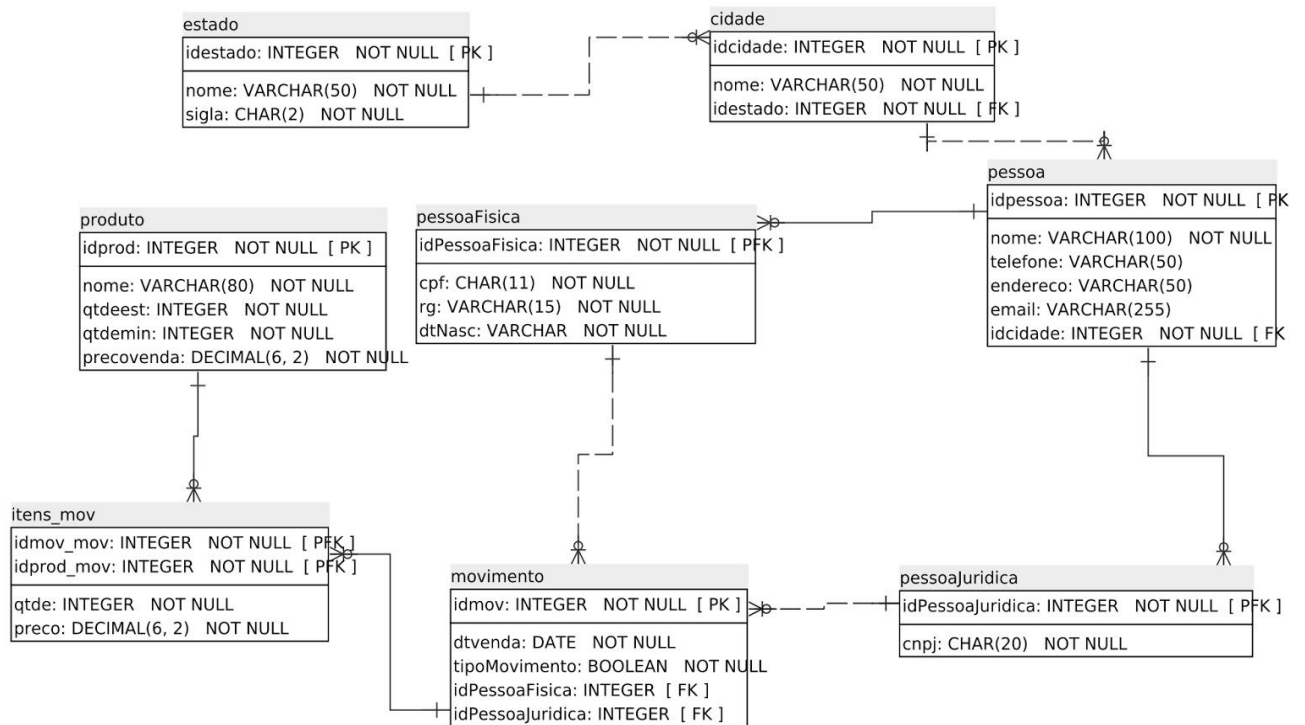
.



Posteriormente foi decidido como iríamos desenvolver o projeto do Banco de dados para trazer maior eficiência juntamente com um sistema de melhor qualidade, para a construção do modelo lógico, desenvolvemos o diagrama entidade relacionamento (DER) e suas cardinalidades de acordo com as funcionalidades estabelecidas devidamente no Canvas, para isso utilizamos o software Power Architect para nos auxiliar na construção, colocamos todas as informações obtidas nas tabelas, exemplificadas na imagem abaixo,



conforme os padrões detectados no Brainstorming, em seguida, exportamos o Dicionário de Dados e o código SQL, que mais tarde importamos no programa PostgreSQL, que pessoalmente é o software de maior eficiência para edição do modelo físico, foram feitas as inserções e testes por meio de comando de consulta utilizando o referido software.



## DICIONÁRIO DE DADOS

### Modelo E-commerce

#### List of tables

- [cidade](#)
- [estado](#)
- [itens\\_mov](#)
- [movimento](#)
- [pessoa](#)
- [pessoaFisica](#)
- [pessoaJuridica](#)
- [produto](#)

#### **cidade** (Physical Name: cidade)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idcidade (PK)	idcidade	INTEGER	PK	NOT NULL
nome	nome	VARCHAR(50)		NOT NULL
idestado ( <a href="#">FK</a> )	idestado	INTEGER		NOT NULL
<b>References</b>				
<ul style="list-style-type: none"> <li>• <a href="#">estado</a> through (idestado)</li> </ul>				
<b>Referenced By</b>				
<ul style="list-style-type: none"> <li>• <a href="#">pessoa</a> referencing (idcidade)</li> </ul>				

#### **estado** (Physical Name: estado)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idestado (PK)	idestado	INTEGER	PK	NOT NULL
nome	nome	VARCHAR(50)		NOT NULL
sigla	sigla	CHAR		NOT NULL
<b>Referenced By</b>				
<ul style="list-style-type: none"> <li>• <a href="#">cidade</a> referencing (idestado)</li> </ul>				

#### **itens\_mov** (Physical Name: itens\_mov)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idmov_mov (PK) ( <a href="#">FK</a> )	idmov_mov	INTEGER	PK	NOT NULL
idprod_mov (PK) ( <a href="#">FK</a> )	idprod_mov	INTEGER	PK	NOT NULL
qtde	qtde	INTEGER		NOT NULL
preco	preco	DECIMAL(6,2)		NOT NULL
<b>References</b>				
<ul style="list-style-type: none"> <li>• <a href="#">movimento</a> through (idmov_mov)</li> <li>• <a href="#">produto</a> through (idprod_mov)</li> </ul>				

#### **movimento** (Physical Name: movimento)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idmov (PK)	idmov	INTEGER	PK	NOT NULL
dtvenda	dtvenda	DATE		NOT NULL
tipoMovimento	tipoMovimento	BOOLEAN		NOT NULL
idPessoaFisica ( <a href="#">FK</a> )	idPessoaFisica	INTEGER		
idPessoaJuridica ( <a href="#">FK</a> )	idPessoaJuridica	INTEGER		
<b>References</b>				
<ul style="list-style-type: none"> <li>• <a href="#">pessoaFisica</a> through (idPessoaFisica)</li> <li>• <a href="#">pessoaJuridica</a> through (idPessoaJuridica)</li> </ul>				
<b>Referenced By</b>				
<ul style="list-style-type: none"> <li>• <a href="#">itens_mov</a> referencing (idmov)</li> </ul>				

**pessoa** (Physical Name: pessoa)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idpessoa (PK)	idpessoa	INTEGER	PK	NOT NULL
nome	nome	VARCHAR(100)		NOT NULL
telefone	telefone	VARCHAR(50)		
endereco	endereco	VARCHAR(50)		
email	email	VARCHAR(255)		
idcidade (FK)	idcidade	INTEGER		NOT NULL
<b>References</b>				
<ul style="list-style-type: none"> <li><a href="#">cidade</a> through (idcidade)</li> </ul>				
<b>Referenced By</b>				
<ul style="list-style-type: none"> <li><a href="#">pessoaFisica</a> referencing (idpessoa)</li> <li><a href="#">pessoaJuridica</a> referencing (idpessoa)</li> </ul>				

**pessoaFisica** (Physical Name: pessoaFisica)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idPessoaFisica (PK) (FK)	idPessoaFisica	INTEGER	PK	NOT NULL
cpf	cpf	CHAR		NOT NULL
rg	rg	VARCHAR(15)		NOT NULL
dtNasc	dtNasc	VARCHAR(0)		NOT NULL
<b>References</b>				
<ul style="list-style-type: none"> <li><a href="#">pessoa</a> through (idPessoaFisica)</li> </ul>				
<b>Referenced By</b>				
<ul style="list-style-type: none"> <li><a href="#">movimento</a> referencing (idPessoaFisica)</li> </ul>				

**pessoaJuridica** (Physical Name: pessoaJuridica)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idPessoaJuridica (PK) (FK)	idPessoaJuridica	INTEGER	PK	NOT NULL
cnpj	cnpj	CHAR		NOT NULL
<b>References</b>				
<ul style="list-style-type: none"> <li><a href="#">pessoa</a> through (idPessoaJuridica)</li> </ul>				
<b>Referenced By</b>				
<ul style="list-style-type: none"> <li><a href="#">movimento</a> referencing (idPessoaJuridica)</li> </ul>				

**produto** (Physical Name: produto)

Logical Column Name	Physical Column Name	Type	PK	Nullable
idprod (PK)	idprod	INTEGER	PK	NOT NULL
nome	nome	VARCHAR(80)		NOT NULL
qtdeest	qtdeest	INTEGER		NOT NULL
qtdemin	qtdemin	INTEGER		NOT NULL
precovenda	precovenda	DECIMAL(6,2)		NOT NULL
<b>Referenced By</b>				
<ul style="list-style-type: none"> <li><a href="#">itens_mov</a> referencing (idprod)</li> </ul>				

## Código SQL - Construção de Tabelas

```

CREATE SEQUENCE public.produto_idprod_seq;

CREATE TABLE public.produto (
    idprod INTEGER NOT NULL DEFAULT nextval('public.produto_idprod_seq'),
    nome VARCHAR(80) NOT NULL,
    qtdeest INTEGER NOT NULL,
    qtdemin INTEGER NOT NULL,
    precovenda NUMERIC(6,2) NOT NULL,
    CONSTRAINT idprod PRIMARY KEY (idprod)
);

ALTER SEQUENCE public.produto_idprod_seq OWNED BY public.produto.idprod;

CREATE SEQUENCE public.estado_idestado_seq_1;

CREATE TABLE public.estado (
    idestado INTEGER NOT NULL DEFAULT nextval('public.estado_idestado_seq_1'),
    nome VARCHAR(50) NOT NULL,
    sigla CHAR(2) NOT NULL,
    CONSTRAINT idestado PRIMARY KEY (idestado)
);

ALTER SEQUENCE public.estado_idestado_seq_1 OWNED BY public.estado.idestado;

CREATE SEQUENCE public.cidade_idcidade_seq;

CREATE TABLE public.cidade (
    idcidade INTEGER NOT NULL DEFAULT nextval('public.cidade_idcidade_seq'),
    nome VARCHAR(50) NOT NULL,
    idestado INTEGER NOT NULL,
    CONSTRAINT idcidade PRIMARY KEY (idcidade)
);

ALTER SEQUENCE public.cidade_idcidade_seq OWNED BY public.cidade.idcidade;

CREATE SEQUENCE public.pessoa_idpessoa_seq;

CREATE TABLE public.pessoa (
    idpessoa INTEGER NOT NULL DEFAULT nextval('public.pessoa_idpessoa_seq'),
    nome VARCHAR(100) NOT NULL,
    telefone VARCHAR(50),
    endereco VARCHAR(50),
    email VARCHAR(255),
    idcidade INTEGER NOT NULL,
    CONSTRAINT idpessoa PRIMARY KEY (idpessoa)
);

ALTER SEQUENCE public.pessoa_idpessoa_seq OWNED BY public.pessoa.idpessoa;

CREATE TABLE public.pessoaJuridica (
    idPessoaJuridica INTEGER NOT NULL,
    cnpj CHAR(20) NOT NULL,
    CONSTRAINT idpessoajuridica PRIMARY KEY (idPessoaJuridica)
);

```



```

CREATE TABLE public.pessoaFisica (
    idPessoaFisica INTEGER NOT NULL,
    cpf CHAR(11) NOT NULL,
    rg VARCHAR(15) NOT NULL,
    dtNasc VARCHAR NOT NULL,
    CONSTRAINT idpessoafisica PRIMARY KEY (idPessoaFisica)
);

CREATE SEQUENCE public.movimento_idmov_seq;

CREATE TABLE public.movimento (
    idmov INTEGER NOT NULL DEFAULT nextval('public.movimento_idmov_seq'),
    dtvenda DATE NOT NULL,
    tipoMovimento BOOLEAN NOT NULL,
    idPessoaFisica INTEGER,
    idPessoaJuridica INTEGER,
    CONSTRAINT idmov PRIMARY KEY (idmov)
);

ALTER SEQUENCE public.movimento_idmov_seq OWNED BY public.movimento.idmov;

CREATE TABLE public.itens_mov (
    idmov_mov INTEGER NOT NULL,
    idprod_mov INTEGER NOT NULL,
    qtde INTEGER NOT NULL,
    preco NUMERIC(6,2) NOT NULL,
    CONSTRAINT iditens_mov PRIMARY KEY (idmov_mov, idprod_mov)
);

ALTER TABLE public.itens_mov ADD CONSTRAINT idprod_itens_venda_fk
FOREIGN KEY (idprod_mov)
REFERENCES public.produto (idprod)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.cidade ADD CONSTRAINT estado_cidade_fk
FOREIGN KEY (idestado)
REFERENCES public.estado (idestado)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.pessoa ADD CONSTRAINT cidade_pessoa_fk
FOREIGN KEY (idcidade)
REFERENCES public.cidade (idcidade)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.pessoaFisica ADD CONSTRAINT pessoa_pessoaFisica_fk
FOREIGN KEY (idPessoaFisica)
REFERENCES public.pessoa (idpessoa)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

```

```
ALTER TABLE public.pessoaJuridica ADD CONSTRAINT pessoa_pessoaJuridica_fk  
FOREIGN KEY (idPessoaJuridica)  
REFERENCES public.pessoa (idpessoa)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE public.movimento ADD CONSTRAINT pessoaJuridica_movimento_fk  
FOREIGN KEY (idPessoaJuridica)  
REFERENCES public.pessoaJuridica (idPessoaJuridica)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE public.movimento ADD CONSTRAINT pessoaFisica_idvend_fk  
FOREIGN KEY (idPessoaFisica)  
REFERENCES public.pessoaFisica (idPessoaFisica)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE public.itens_mov ADD CONSTRAINT idvend_itens_venda_fk  
FOREIGN KEY (idmov_mov)  
REFERENCES public.movimento (idmov)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

### Código SQL - Inserções

```
INSERT INTO estado(nome, sigla) VALUES
('PARANÁ','PR'),
('SÃO PAULO','SP'),
('RIO DE JANEIRO','RJ');
```

```
INSERT INTO cidade(nome,idestado) VALUES
('Maringá', 1),
('Paranavaí', 1),
('Marilena', 1),
('Campinas', 2),
('Petrópolis', 3);
```

```
INSERT INTO pessoa(nome,telefone, endereco, email, idcidade) VALUES
('Info & cia', 'Rua das Nações','34421515','beer@gmail.com',5),
('Escritório Atimus', 'Rua do Irineu','34421515','bebidasCia@gmail.com',5),
('Rede Livre Comércio', 'Rua Groove', '34221190','diversity@gmail.com',4),
('José Amado', 'Rua do ABC','30451120', 'amador@hotmail.com',1),
('Ana Maria', 'Rua João da Silva', '30456711','anamaria@outlook.com',1),
('João Desah Parecido', 'Avenida do Caneco','31459212', 'jao123@yahoo.com',2),
('Bruce Li', 'Rodovia BR 376', '34239763','libruce@hotmail.com',3),
('Josefina Lima', 'Rua Almirante', '34231121', 'jose@gmail.com', 1),
('Abrilina Décima', 'Rua do Nunca','34228712','abrillna@gmail.com',4),
('Amin Amou', 'Avenida Parigot', '31458231','amado@yahoo.com.br', 2),
('Jacinto Leite', 'Rua do Amanhã', '34239812','jacin123@gmail.com',1),
('Manoel Igualdade', 'Rua Amampá', '34231212','mano@gmail.com',1);
```

```
INSERT INTO pessoaFisica(idPessoaFisica, cpf, rg, dtNasc) VALUES
(4,'81459143027','448806228','1998-12-11'),
(5,'94202065050','267840858','1982-05-10'),
(6,'16160812017','378112247','1970-04-09'),
(7,'02765336075','103573355','1998-11-03'),
(8,'53300858020','497873643','1995-01-02'),
(9,'51583614010','225069684','1994-02-06'),
(10,'49924003020','224492731','1930-11-05'),
(11,'98962117010','470223832','2002-04-12'),
(12,'97809538098','419428884','1940-12-12');
```

```
INSERT INTO pessoaJuridica(idpessoaJuridica, cnpj) VALUES
(1,'43.541.228/0001-00'),
(2,'35.763.037/0001-56'),
(3,'30.051.110/0001-43');
```



```

INSERT INTO movimento(dtvenda, tipoMovimento, idPessoaFisica, idPessoaJuridica) VALUES
/*True é uma venda, False é uma compra*/
('2019-11-20', True ,NULL,1), /*ULTIMO CAMPO FOI UMA COMPRA/VENDA DO pessoaJuridica*/
('2019-11-21', False ,NULL,1),
('2019-11-25', True ,NULL,2),
('2019-11-20', False ,NULL,2),
('2019-11-21', False ,NULL,3),
('2019-11-25', False ,NULL,3),
('2019-05-02', True ,4,NULL), /*PENULTIMA FOI UMA VENDA PARA pessoaFisica*/
('2019-11-04', True ,4,NULL),
('2019-11-05', True ,5,NULL),
('2019-11-04', True ,5,NULL),
('2019-11-05', True ,6,NULL),
('2019-11-04', True ,6,NULL);

INSERT INTO produto (nome, qtdeest, qtdemin, precovenda) /*Alterar produtos*/
VALUES ('Notebook Lenovo i3', 40, 20, 1900.00),
('Notebook Lenovo Quadcore', 30, 20, 400.50),
('HD externo 1 TB', 48, 40, 224.50),
('HD externo 2 TB', 60, 40, 355.50),
('HD externo 4 TB', 48, 40, 529.50),
('Monitor Acer HD', 20, 12, 1100.00),
('Pendrive 4 gb', 48, 25, 20.00),
('Pendrive 8 gb', 48, 25, 40.50),
('Pendrive 16 gb', 48, 25, 60.00),
('Iphone 7', 20, 10, 2900.00),
('Moto g7 play', 40, 25, 999.99),
('Moto g7', 40, 25, 1200.50),
('Headset Gamer Razer Kraken', 20, 12, 521.50),

('Fone de ouvido Intra Auricular', 48, 25, 40.50),
('Mouse sem fio', 40, 20, 40.00),
('Mouse Hardline', 40, 30, 15.00);

/*alterar tudo isso*/
INSERT INTO itens_mov(idmov_mov,idprod_mov,qtde,preco) VALUES
(1,1,3,'1900.00'), /*Atenção repita o preço do produto, pois ficará no histórico,
(1,2,4,'400.50'),
(2,3,5,'224.50'),
(2,4,6,'355.50'),
(3,5,7,'529.50'),
(3,6,4,'1100.00'),
(4,7,2,'20.00'),
(4,8,2,'40.50'),
(5,9,2,'60.00'),
(5,10,2,'2900.00'),
(6,11,2,'999.99'),
(6,12,2,'1200.50');

```



## Código SQL - Selects

```
/*Fazer um balanço financeiro MENSALMENTE para ter uma noção das DESPESAS em compras com o pessoaJuridica.*/
SELECT pessoa.nome"Nome da Empresa Juridica", movimento.dtvenda"Data da Compra", movimento.tipoMovimento,
       (itens_mov.preco*qtde)"Preço Total", itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa INNER JOIN pessoaJuridica
ON pessoa.idpessoa = pessoaJuridica.idPessoaJuridica
INNER JOIN movimento
ON pessoaJuridica.idPessoaJuridica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod
WHERE dtvenda BETWEEN '2019-11-01' AND '2019-11-30' AND tipomovimento = False;
```

```
/*Fazer um balanço financeiro MENSALMENTE para se ter uma noção da margem de lucro com Pessoas Físicas.*/
SELECT pessoa.nome"Nome da pessoa fisica", movimento.dtvenda"Data da Venda", (itens_mov.preco*qtde)"Preço",
       itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa
LEFT JOIN pessoaFisica
ON pessoa.idpessoa = pessoaFisica.idPessoaFisica
INNER JOIN movimento
ON pessoaFisica.idPessoaFisica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod
```

```
/*Fazer um balanço financeiro MENSALMENTE para se ter uma noção da margem de lucro com Pessoas Juridicas.*/
SELECT pessoa.nome"Nome da pessoa juridica", movimento.dtvenda"Data da Venda", movimento.tipoMovimento,
       (itens_mov.preco*qtde)"Preço", itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa
INNER JOIN pessoaJuridica
ON pessoa.idpessoa = pessoaJuridica.idPessoaJuridica
INNER JOIN movimento
ON pessoaJuridica.idPessoaJuridica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod
WHERE dtvenda BETWEEN '2019-11-01' AND '2019-11-30' AND tipoMovimento = True ;
```

```

/*Fazer um balanço financeiro DIARIAMENTE para ter um controle sobre as receitas com Pessoas Fisicas*/
SELECT pessoa.nome"Nome da Pessoa Fisica", movimento.dtvenda"Data da Venda", (itens_mov.preco*qtde)"Preço",
       itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa INNER JOIN pessoaFisica
ON pessoa.idpessoa = pessoaFisica.idPessoaFisica
INNER JOIN movimento
ON pessoaFisica.idPessoaFisica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod
WHERE dtvenda BETWEEN '2019-11-25' AND '2019-11-25';

/*Fazer um balanço financeiro DIARIAMENTE para ter um controle sobre as receitas com Pessoas Juridicas*/
SELECT pessoa.nome"Nome da Pessoa Fisica", movimento.dtvenda"Data da Venda", movimento.tipoMovimento, (itens_mov.preco*qtde)"Preço",
       itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa INNER JOIN pessoaJuridica
ON pessoa.idpessoa = pessoaJuridica.idPessoaJuridica
INNER JOIN movimento
ON pessoaJuridica.idPessoaJuridica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod
WHERE dtvenda BETWEEN '2019-11-25' AND '2019-11-25' AND tipoMovimento = True;

/*Fazer um balanço financeiro DIARIAMENTE para ter um controle sobre as despesas*/
SELECT pessoa.nome"Nome da PessoaJuridica", movimento.dtvenda"Data da Compra", movimento.tipoMovimento,
       (itens_mov.preco*qtde)"Preço Total", itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa INNER JOIN pessoaJuridica
ON pessoa.idpessoa = pessoaJuridica.idPessoaJuridica
INNER JOIN movimento
ON pessoaJuridica.idPessoaJuridica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod
WHERE dtvenda BETWEEN '2019-11-21' AND '2019-11-21' AND tipoMovimento = False; /*colocar itens */

/*Relação de todos os clientes -vendas- histórico*/
SELECT pessoa.nome"Nome do Cliente", movimento.dtvenda"Data da Venda",
       (itens_mov.preco*qtde)"Preço", itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa INNER JOIN pessoaFisica
ON pessoa.idpessoa = pessoaFisica.idPessoaFisica
INNER JOIN movimento
ON pessoaFisica.idPessoaFisica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod;

/*Relação de todas as Pessoas Juridicas -Compras- histórico*/
SELECT pessoa.nome"Nome da Pessoa Juridica", movimento.dtvenda"Data da Compra",
       (itens_mov.preco*qtde)"Preço Total", itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa INNER JOIN pessoaJuridica
ON pessoa.idpessoa = pessoaJuridica.idPessoaJuridica
INNER JOIN movimento
ON pessoaJuridica.idPessoaJuridica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod AND tipoMovimento = False;

```

```

/*Relação de todos as Pessoas Juridicas -Vendas- histórico*/
SELECT pessoa.nome"Nome da Pessoa Juridica", movimento.dtvenda"Data da Compra",
       (itens_mov.preco*qtde)"Preço Total", itens_mov.qtde"Quantidade", produto.nome"Nome do Produto"
FROM pessoa INNER JOIN pessoaJuridica
ON pessoa.idpessoa = pessoaJuridica.idPessoaJuridica
INNER JOIN movimento
ON pessoaJuridica.idPessoaJuridica = movimento.idmov
INNER JOIN itens_mov
ON movimento.idmov = itens_mov.idmov_mov
INNER JOIN produto
ON itens_mov.idprod_mov = produto.idprod AND tipoMovimento = True;

/*Permitir a busca pela quantidade do produto em estoque, para se ter noção dos produtos armazenados*/
SELECT nome"Nome do Produto", qtdeEst"Quantidade em estoque" FROM produto;

/*Viabilizar a consulta por um determinado produto, para possíveis alterações futuras. */
SELECT nome"Nome do Produto", qtdeest"Quantidade em estoque", qtdemin"Quantidade Mínima", precovenda"Preço" FROM produto
WHERE nome LIKE '%H%';

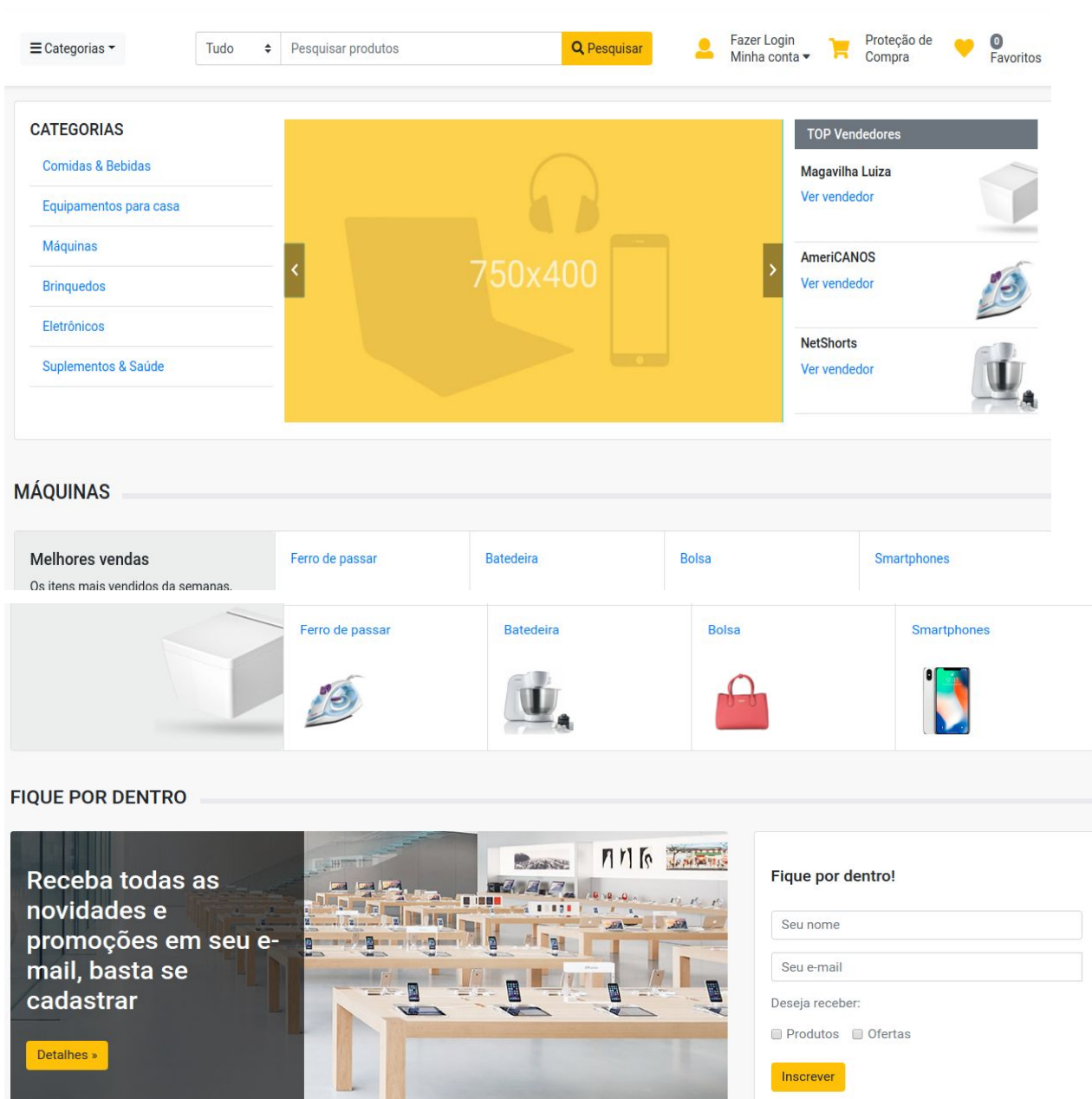
/*Permitir fazer a busca de fornecedores pelo nome, para possíveis alterações futuras*/
SELECT pessoa.nome"Nome da PessoaJuridica", pessoa.telefone"Telefone", pessoa.endereco"Endereço",
       pessoa.email"Email", pessoaJuridica.cnpj"CNPJ", cidade.nome"Cidade", estado.nome"Estado"
FROM pessoa INNER JOIN pessoaJuridica
ON pessoa.idpessoa = pessoaJuridica.idPessoaJuridica
INNER JOIN cidade
ON pessoa.idcidade = cidade.idcidade
INNER JOIN estado
ON estado.idestado = cidade.idestado
WHERE pessoa.nome LIKE '%cia%';

/*Permitir fazer a busca de clientes pelo nome, para possíveis alterações futuras*/
SELECT pessoa.nome"Nome da Pessoa", pessoa.telefone"Telefone", pessoa.endereco"Endereço", pessoa.email"Email",
       pessoaFisica.cpf"CPF", pessoaFisica.rg"RG", pessoaFisica.dtNasc"Data de Nasc", cidade.nome"Cidade", estado.nome"Estado"
FROM pessoa INNER JOIN pessoaFisica
ON pessoa.idpessoa = pessoaFisica.idPessoaFisica
INNER JOIN cidade
ON pessoa.idcidade = cidade.idcidade
INNER JOIN estado
ON estado.idestado = cidade.idestado
WHERE pessoa.nome LIKE '%Ana%';













```



## Prototipação:



## ITENS RECOMENDADOS

 <p>Bolsa R\$ 500,00 R\$1.200,00</p>	 <p>iPhone X R\$ 3.000,00 R\$4.000,00</p>	 <p>Sofá Tech R\$ 280,00</p>	 <p>Poltrona Luxuosa R\$ 300,00</p>	 <p>Notebok AlienWare 15 R\$ 22.000,00 R\$ 23.000,00</p>	 <p>Notebook AlienWare M14X R\$ 14.000,00</p>
 <p>4K</p>		 <p>WHEY</p>	 <p>BCAA 2400</p>		

## Fazer login

Preencha com suas informações

## Crie sua conta

Preencha com suas informações

Senha

Você é;

Pessoa Física

Criar conta

### 3 CONCLUSÃO

Ao finalizarmos o projeto, notamos como é importante a boa elaboração de um banco de dados, já que seu bom funcionamento pode ajudar enormemente no gerenciamento de dados. Porém, erros na construção de sua estrutura podem levar uma empresa a falência, outro destaque é a fase de coleta de requisitos, ela é extremamente importante e indispensável, é mais importante gastar tempo em uma boa coleta de requisitos, do que na construção do projeto em si, já que uma vez que o software é criado se torna muito difícil implementar alterações, basicamente isso foi aprendido na prática, além de aumentar o preço do desenvolvimento do produto final.

#### **Análise de Custo do Sistema:**

Análise do Projeto: 5 horas

Elaboração do cadastro de clientes: 12 horas

Elaboração do cadastro de produtos: 12 horas

Elaboração de rotinas de pagamento: 6 horas

Alterar clientes/Excluir Clientes: 2 horas

Desenvolvimento de rotinas: 9 horas

Desenvolvimento do fluxo de caixa: 12 horas

Desenvolvimento de relatórios: 14 horas

Elaboração de requisitos: 12 horas

Valor da hora: R\$ 78,00

Sendo assim, foi feita uma jornada de 84 horas, resultando no valor final do projeto em R\$1008,00