

UNIVERSIDADE DO OESTE DE SANTA CATARINA
ÁREA DAS CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ARTHUR SCHMIDT
LUCAS BUENO
LUIS AUGUSTO BERWNAGER MALLMANN
NATAN VIDORI

CARNESHOP

SÃO MIGUEL DO OESTE - SC
2018
ARTHUR SCHMIDT
LUCAS BUENO

LUIS AUGUSTO BERWNAGER MALLMANN
NATAN VIDORI

CARNESHOP

Trabalho da disciplina de Banco de Dados I e Estrutura de Dados II apresentado ao Curso de Ciência da Computação, Área das Ciências Exatas e Tecnológicas, da Universidade do Oeste de Santa Catarina – Campus de São Miguel do Oeste – SC, como requisito para composição de média final.

Orientador: Prof. Msc. Roberson Junior Fernandes Alves

SÃO MIGUEL DO OESTE - SC
2018
LISTA DE ILUSTRAÇÕES

Figura 1: Modelo Entidade-Relacionamento CarneShop	6
Figura 2: Código PHP(PDO) de Conexão do ao Banco de Dados	7
Figura 3: Tela inicial apresentada ao cliente	8
Figura 4: Painel de Controle do Sistema - Tela do Administrador	9
Figura 5: Nova tela inicial	9

SUMÁRIO

1 INTRODUÇÃO	4
--------------------	---

2	CARNESHOP	5
2.1	LISTA DE REQUISITOS	5
2.2	MODELO ENTIDADE-RELACIONAMENTO (BANCO DE DADOS)	6
2.3	BANCO DE DADOS	6
2.3.1	SCRIPTS DO BANCO DE DADOS	6
2.3.2	SGBD – POSTGRESSQL	7
2.3.3	INTEGRAÇÃO BANCO DE DADOS E SISTEMA	7
2.4	DESENVOLVIMENTO WEB.....	8
2.4.1	INTERFACE DE CLIENTE	8
2.4.2	INTERFACE DE ADMINISTRADOR.....	9
2.5	RELATÓRIOS	9
3	CONCLUSÃO	11
	REFERÊNCIAS	12
	ANEXOS	13

1 INTRODUÇÃO

Nos últimos anos as mais variadas atividades passaram ou iniciaram o processo de adaptação de suas metodologias de trabalho tradicionais para novos modelos compatíveis com a atual era digital. No meio comercial/varejista não foi diferente. Através do crescimento das transações no comércio eletrônico inúmeras empresas apostaram nesse segmento e conseguiram melhorar seus resultados, visto que, hoje esse tipo de comércio corresponde a uma grande parcela do mercado brasileiro e mundial.

Buscando aproveitar essa tendência e a existência de nichos de mercado ainda pouco explorados foi desenvolvido o CarneShop, sistema de comércio eletrônico de carnes para churrasco.

Disponibilizado em interface web, o sistema permite que o usuário escolha as carnes desejadas e solicite a entrega em seu endereço. Atualmente ainda em desenvolvimento, aceita pagamentos somente em dinheiro, no momento da entrega do produto.

2 CARNESHOP – Versão 1/2018

2.1 LISTA DE REQUISITOS

Inicialmente foram levantados alguns requisitos funcionais que o sistema deveria atender. Durante o desenvolvimento, foram atendidos a maioria dos requisitos estabelecidos. Os demais ainda podem ser desenvolvidos conforme a evolução e aprimoramento do sistema.

N.	Descrição do Requisito*
1	O sistema deve cadastrar o cliente com os seus dados, exemplo: nome, cpf/cnpj, endereço, telefone, email
2	O sistema deve validar o cpf e o cep (endereço)
3	O sistema deve apresentar controle de estoque (gerar relatório)
4	O sistema deve apresentar uma interface para cadastramento de produtos
5	O sistema deve ser seguro (validação do cadastro do usuário via email, criptografia de senhas, Validação de cartões de crédito)
6	O sistema deve apresentar o histórico de pedidos de cada usuário (<u>tanto</u> para uso do usuário quanto para uso interno)
7	O sistema deve ser capaz de organizar os produtos por categorias
8	O sistema deve apresentar um buscador de produtos
9	O sistema deve apresentar uma página de informações para contato
10	O sistema deve no cadastramento de produtos permitir informar a descrição e fotos
12	Para realizar compras o usuário deve Realizar login
13	O sistema deve apresentar uma plataforma para envio de newsletter para os e-mails cadastrados
14	O sistema deve oferecer acompanhamento da compra (site dos correios/transportadora)
15	O sistema deve destacar para o usuários produtos em destaque/promoção

*** Requisito: Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos.**

2.2 MODELO ENTIDADE-RELACIONAMENTO (BANCO DE DADOS)

A partir da análise dos requisitos levantados e com a finalidade definida foi construído o modelo de entidade-relacionamento do banco de dados do sistema através da ferramenta CASE Visual Paradigm. O modelo foi construído de forma simples, mas o suficiente para atender as funcionalidades desejadas.

Durante a modelagem, foram efetuados comentários e descrições relativos a cada tabela e atributos criados para possibilitar a compreensão do modelo por terceiros e também para a criação de um dicionário de dados (em anexo).

O modelo construído é seguinte:

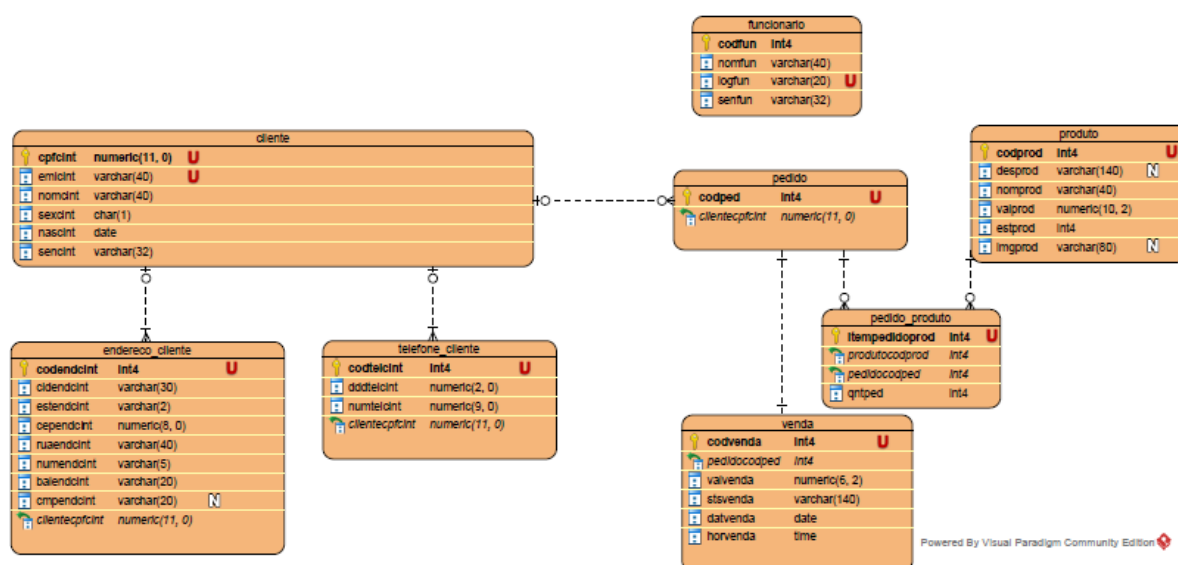


Figura 1: Modelo Entidade-Relacionamento CarneShop

Fonte: Autores

2.3 BANCO DE DADOS

2.3.1 SCRIPTS DO BANCO DE DADOS

Com o modelo entidade-relacionamento criado, a base para o do banco de dados estava pronta. Utilizando novamente o Virtual Paradigm e o driver disponível para o SGBD (Sistema Gerenciador de Banco de Dados) PostgreSQL, foram criados os scripts para a criar e apagar as tabelas de dados do banco.

Os scripts gerados podem ser visualizados nos anexos B e C deste trabalho.

2.3.2 SGBD – POSTGRESQL

O gerenciamento do banco de dados do CarneShop foi realizado com a utilização do PostgreSQL. Ele é um Sistema Gerenciador de Banco de Dados de código aberto e multiplataforma, desenvolvido pela PostgreSQL Global Development Group. Ele conta com inúmeros recursos, como consultas complexas, visões, controle de concorrência, chaves estrangeiras, entre outros.

A ferramenta PgAdmin foi utilizada em conjunto, visto que, oferece um ambiente gráfico para a administração do SGBD PostgreSQL.

2.3.3 INTEGRAÇÃO BANCO DE DADOS E SISTEMA

A integração do banco de dados com o sistema web desenvolvido é realizada pela linguagem de programação PHP, através do seguinte código:

```

1  <?php
2  try {
3      // cria o objeto de conexão PDO
4      $conexao = new PDO('pgsql:host=localhost;dbname=carneshop',
5                          // usuario, senha
6                          'postgres', 'postgres');
7      $conexao->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
8  } catch (PDOException $e) {
9      echo $e->getMessage();
10 }
11 ?>

```

Figura 2: Código PHP(PDO) de Conexão do ao Banco de Dados
Fonte: Autores

A integração é realizada com a utilização da biblioteca do PHP chamada de PDO – PHP Data Object, facilitando o processo de conexão e comunicação com o banco de dados, visto que, esta biblioteca cria uma camada abstrata e efetua a conexão com diversos bancos de dados da mesma maneira, modificando apenas a sua string de conexão.

2.4 DESENVOLVIMENTO WEB

O desenvolvimento da interface web foi realizado através da implementação dos códigos utilizando a estrutura HTML, CSS, JavaScript e PHP.

O CarneShop pode ser dividido em duas interfaces: a de cliente, interface que apresenta ao usuário o sistema de compra; e a de funcionário, interface de uso restrito aos funcionários e administradores que apresenta ferramentas de controle e gerenciamento do sistema CarneShop como um todo.

A seguir serão apresentadas algumas das imagens das interfaces criadas:

2.4.1 INTERFACE DE CLIENTE

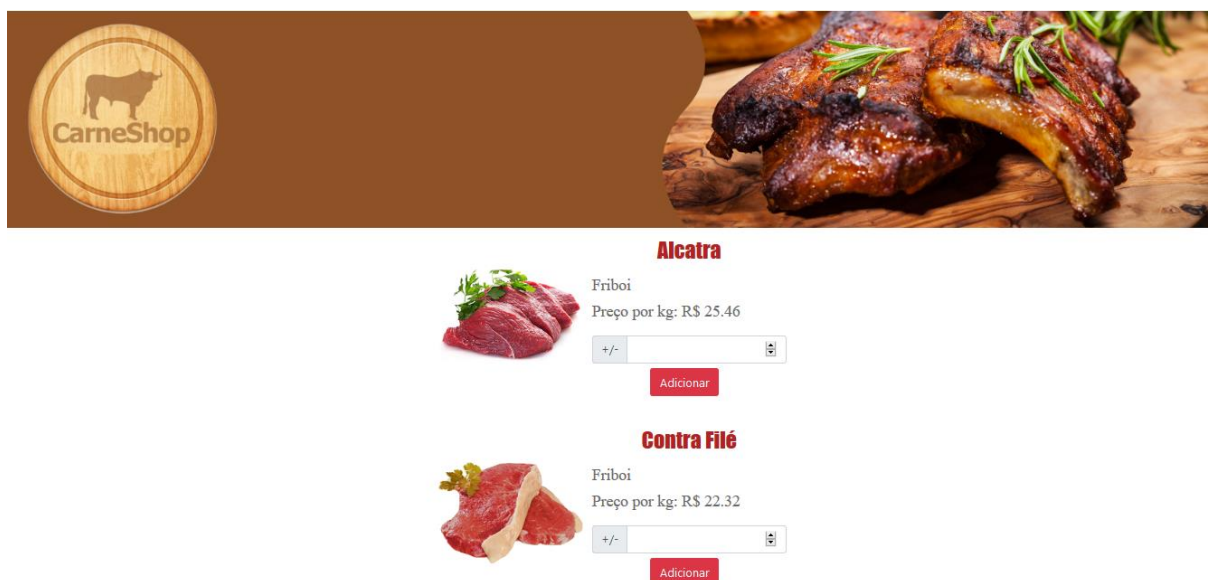


Figura 3: Tela inicial apresentada ao cliente
Fonte: Autores

2.4.2 INTERFACE DE ADMINISTRADOR



Figura 4: Painel de Controle do Sistema - Tela do Administrador
Fonte: Autores

2.5 RELATÓRIOS

Como o desenvolvimento desse projeto é requisito para composição de média final das disciplinas de Banco de Dados I e Estrutura de Dados II, um dos itens avaliados é a apresentação de relatórios

a partir da base de dados criada. Os relatórios a serem apresentados foram definidos pelo professor com base no modelo de entidade-relacionamento do banco.

Sendo assim, os relatórios a serem gerados e seus respectivos scripts de consulta ao banco são os seguintes:

- a) Relacionar o código e nome de clientes do sexo feminino com idades entre 20 e 30 anos da cidade de Itapiranga - SC. Ordenar o relatório em ordem alfabética.

```
SELECT C.CPFCLNT, C.NOMCLNT FROM CLIENTE C
INNER JOIN ENDERECO_CLIENTE T ON C.CPFCLNT = T.CLIENTECPFCLNT
WHERE C.SEXCLNT = 'F' AND T.CIDENDCLNT ILIKE 'ITAPIRANGA' AND
(CAST(TO_CHAR(AGE(CURRENT_DATE, C.NASCLNT), 'YY') AS INTEGER) >= 20
AND CAST(TO_CHAR(AGE(CURRENT_DATE, C.NASCLNT), 'YY') AS INTEGER) <=
30)
ORDER BY C.NOMCLNT;
```

- b) Relacionar o código e nome de clientes com mais de 2 telefones cadastrados.

```
SELECT C.CPFCLNT, C.NOMCLNT FROM CLIENTE C
INNER JOIN TELEFONE_CLIENTE T ON C.CPFCLNT = T.CLIENTECPFCLNT
GROUP BY C.CPFCLNT
HAVING COUNT(T.CODTELCLNT) >= 2;
```

- c) Relacionar código, quantidade e valor total, agrupadas por mês de vendas realizadas em meses pares de 2017. Relacionar da venda com maior valor para a venda com menor valor.

```
SELECT DATE_PART('MONTH' , DATVENDA) AS MÊS, V.CODVENDA,
COUNT(E.PRODUTOCODPROD) AS "QNT DE ITENS", V.VALVENDA FROM
VENDA V
INNER JOIN PEDIDO D ON V.PEDIDOCODPED = D.CODPED
INNER JOIN PEDIDO_PRODUTO E ON D.CODPED = E.PEDIDOCODPED WHERE
(CAST(DATE_PART('MONTH' , DATVENDA) AS INTEGER)%2 = 0) AND
(DATE_PART('YEAR' , DATVENDA) = '2017')
GROUP BY V.CODVENDA ORDER BY DATE_PART('MONTH' , DATVENDA),
VALVENDA DESC;
```

d) Relacionar o código e nome do produto, data e valor total da última venda realizada. Ordenar o relatório em ordem alfabética.

```
SELECT p.codprod, p.nomprod, v.datvenda, v.valvenda
FROM pedido_produto pp
INNER JOIN pedido pe ON pe.codped = pp.pedidocodped
INNER JOIN venda v on v.pedidocodped = pe.codped
INNER JOIN produto p on p.codprod = pp.produtocodprod
WHERE v.datvenda = (select max(datvenda) from venda)
and v.horvenda = (select max(horvenda) from venda where datvenda = v.datvenda)
order by p.nomprod;
```

3 CARNESHOP – Versão 2/2018

A segunda versão do projeto do CarneShop contou com uma reformulação da interface inicial mais otimizada, juntamente com a adição de um menu para melhor visualização dos produtos disponíveis no site:

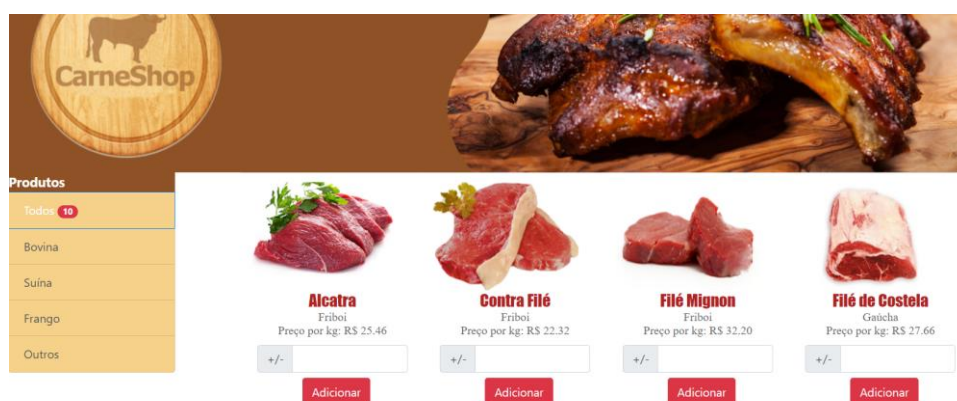


Figura 5: Nova tela inicial

Fonte: Autores

Ajuste nas views e junções o SQL buscando otimizar e facilitar o acesso a informações, bem como o ajuste das restrições (constraints) na criação do banco de dados. Foi configurado as políticas de acesso, criação de usuários e concessão de privilégios no banco de dados.

Foi implementado as políticas de backup, dada a importância das informações armazenadas pelo sistema desenvolvido, optamos por adotar a seguinte estratégia para backup:

1- Backup Lógico da base de dados efetuado diariamente (de preferência durante a madrugada, devido ao horário de menor fluxo de visitantes) através do agendamento do arquivo .bat (disponível nesta mesma pasta);

2 – Backup Físico da base de dados efetuado semanalmente (de preferência durante a madrugada de domingo, devido ao horário de menor fluxo de visitantes);

Para aumentar a disponibilidade do backup, o ideal é manter cópias em um servidor local e outra na nuvem.

Cada backup deve ser armazenado pelo período de um mês e após isso pode ser eliminado.

Além disso foi feita a política de restore. Antes de efetuar qualquer comando de restauração, é necessário verificar a integridade, compatibilidade e confiabilidade dos dados para evitar possíveis problemas e incompatibilidades.

Para finalizar foi otimizado o cadastro dos clientes, com preenchimento validação e preenchimento automático do CEP e validação do CPF. Foi criado o histórico dos produtos e foi implementado a interface gráfica para consulta de relatórios na DB.

4 CONCLUSÃO

O projeto CarneShop buscou a partir da premissa de que a maioria das pessoas são consumidoras de carne e do crescimento constante do comércio eletrônico, criar um sistema web de venda de carnes para churrasco com entrega a domicílio.

Os requisitos funcionais pré-estabelecidos foram em sua maioria atendidos, porém, alguns ainda podem ser implementados durante a evolução e crescimento da aplicação.

Além disso, como o desenvolvimento desse sistema fazia parte dos componentes curriculares de Banco de Dados I e Estrutura de Dados II, a elaboração permitiu que os conteúdos apresentados nas disciplinas fossem compreendidos e aplicados de forma prática.

Dessa forma, o projeto permitiu a partir da criação de uma aplicação real, que com alguns ajustes pode ser lançada ao mercado, a assimilação dos conteúdos teóricos das disciplinas proporcionando maior aproveitamento da disciplina e conhecimento das ferramentas disponíveis para desenvolvimento web, modelagem e gerenciamento de banco de dados.

REFERÊNCIAS


DEVMEDIA. **Introdução ao PHP PDO**. 2012. Disponível em: <<https://www.devmedia.com.br/introducao-ao-php-pdo/24973>>. Acesso em: 24 jun. 2018.





DEVMEDIA. **Introdução ao PostgreSQL**. 2007. Disponível em: <<https://www.devmedia.com.br/introducao-ao-postgresql/6390>>. Acesso em: 24 jun. 2018.


NAKAMURA, André M. **Comércio Eletrônico Riscos nas Compras pela Internet**. 2011. Disponível em: <<http://www.fatecsp.br/dti/tcc/tcc0027.pdf>>. Acesso em: 24 jun. 2018.

ANEXOS

ANEXO A – DICIONÁRIO DE DADOS

Entity Name	Entity Description					
Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
 cliente	Tabela que contém os dados pessoais de cada cliente.					
cpfclnt	CPF do Cliente	numeric	11	true	false	true
emlclnt	Email do Cliente	varchar	40	false	false	true
nasclnt	Data de Nascimento do Cliente	date	0	false	false	false
nomclnt	Nome do Cliente	varchar	40	false	false	false
senclnt	senha de acesso do cliente	varchar	32	false	false	false
sexclnt	sexo do cliente	char	1	false	false	false
 endereco_cliente	Enderecos do cliente. Pode existir mais de um para o mesmo cliente					
baiendclnt	bairro do endereco cadastrado	varchar	20	false	false	false
cependclnt	CEP do endereco cadastrado	numeric	8	false	false	false
cidendclnt	cidade do endereco cadastrado	varchar	30	false	false	false
clientecpfclnt	chave estrangeira que identifica a qual cliente está vinculado o endereco	numeric	11	false	false	false
cmpendclnt	complemento do endereco cadastrado. pode ser opcional	varchar	20	false	true	false
codendclnt	codigo do endereco do cliente. é gerado automaticamente de forma sequencial.	int4	0	true	false	true
estendclnt	estado do endereco cadastrado	varchar	2	false	false	false
numendclnt	numero do endereco	varchar	5	false	false	false
ruaendclnt	rua do endereco cadastrado	varchar	40	false	false	false
 funcionario	detalhamento do funcionario/administrador do sistema					
codfun	código do funcionario, gerado automaticamente de forma sequencial	int4	0	true	false	false

Entity Name	Entity Description					
logfun	login do funcionario	varchar	20	false	false	true
nomfun	nome do funcionario	varchar	40	false	false	false
senfun	senha do funcionario	varchar	32	false	false	false
 pedido	tabela que contem o codigo de identificação do pedido e a qual cliente ele está vinculado. Lembrando que nem todo pedido é uma venda.					
clientecpf clnt	chave estrangeira a qual vincula o cpf ao pedido	numeric	11	false	false	false
codped	codigo do pedido	int4	0	true	false	true
 pedido_pro duto	tabela que relaciona os produtos selecionados ao pedido. pode existir mais de um produto no mesmo pedido					
itempedid oproduct	item do produto no pedido (ex: 1,2,3...)	int4	0	true	false	true
pedidocod ped	chave estrangeira que identifica o pedido	int4	0	false	false	false
produtoco dprod	chave estrangeira que identifica o produto	int4	0	false	false	false
qntped	quantidade de itens pedidos do mesmo produto	int4	0	false	false	false
 produto	informacoes do produto					
codprod	codigo do produto gerado de forma sequencial	int4	0	true	false	true
desprod	descricao do produto	varchar	140	false	true	false
estprod	quantidade em estoque do produto	int4	0	false	false	false
imgprod	nome de imagem cadastrada na pasta imgproduto	varchar	80	false	true	false
nomprod	nome do produto	varchar	40	false	false	false
valprod	valor do produto(R\$)	numeric	10.2	false	false	false
 telefone_cli ente	telefones do cliente. pode existir mais de um registro					
clientecpf clnt	chave estrangeira que identifica a qual cliente está vinculado o telefone	numeric	11	false	false	false
codtelclnt	codigo do telefone do cliente. gerado automaticamente de forma sequencial	int4	0	true	false	true

Entity Name	Entity Description					
dddteleclnt	DDD do telefone do cliente.	numeric	2	false	false	false
numteleclnt	numero do telefone cadastrado para o cliente	numeric	9	false	false	false
 venda	detalhamento da venda (pedido efetuado)					
codvenda	codigo da venda	int4	0	true	false	true
datvenda	data venda	date	0	false	false	false
horvenda	horario da venda	time	0	false	false	false
pedidocod ped	codigo do pedido ao qual a venda está relacionada	int4	0	false	false	false
stsvenda	status da venda	varchar	140	false	false	false
valvenda	valor total da venda	numeric	6.2	false	false	false

ANEXO B – SCRIPT SQL PARA CRIAR TABELAS

```
CREATE TABLE cliente (
```

```

    cpfclnt numeric(11, 0) NOT NULL,
    emlclnt varchar(40) NOT NULL UNIQUE,
    nomclnt varchar(40) NOT NULL,
    sexclnt char(1) NOT NULL,
    nascclnt date NOT NULL,
    sencclnt varchar(32) NOT NULL,
    PRIMARY KEY (cpfclnt));
COMMENT ON TABLE cliente IS 'Tabela que contém os dados
pessoais de cada cliente.';
COMMENT ON COLUMN cliente.cpfclnt IS 'CPF do Cliente';
COMMENT ON COLUMN cliente.emlclnt IS 'Email do Cliente';
COMMENT ON COLUMN cliente.nomclnt IS 'Nome do Cliente';
COMMENT ON COLUMN cliente.sexclnt IS 'sexo do cliente';
COMMENT ON COLUMN cliente.nascclnt IS 'Data de Nascimento do
Cliente';
COMMENT ON COLUMN cliente.sencclnt IS 'senha de acesso do
cliente';
CREATE TABLE endereco_cliente (
    codendclnt      SERIAL NOT NULL,
    cidendclnt      varchar(30) NOT NULL,
    estendclnt      varchar(2) NOT NULL,
    cependclnt      numeric(8, 0) NOT NULL,
    ruaendclnt      varchar(40) NOT NULL,
    numendclnt      varchar(5) NOT NULL,
    baiendclnt      varchar(20) NOT NULL,
    cmpendclnt      varchar(20),
    clientecpfclnt numeric(11, 0) NOT NULL,
    PRIMARY KEY (codendclnt));
COMMENT ON TABLE endereco_cliente IS 'Enderecos do cliente.
Pode existir mais de um para o mesmo cliente';
COMMENT ON COLUMN endereco_cliente.codendclnt IS 'codigo do
endereco do cliente. é gerado automaticamente de forma
sequencial.';
COMMENT ON COLUMN endereco_cliente.cidendclnt IS 'cidade do
endereco cadastrado';
COMMENT ON COLUMN endereco_cliente.estendclnt IS 'estado do
endereco cadastrado';
COMMENT ON COLUMN endereco_cliente.cependclnt IS 'CEP do
endereco cadastrado';
COMMENT ON COLUMN endereco_cliente.ruaendclnt IS 'rua do
endereco cadastrado';
COMMENT ON COLUMN endereco_cliente.numendclnt IS 'numero do
endereco';
COMMENT ON COLUMN endereco_cliente.baiendclnt IS 'bairro do
endereco cadastrado';
COMMENT ON COLUMN endereco_cliente.cmpendclnt IS 'complemento
do endereco cadastrado. pode ser opcional';
COMMENT ON COLUMN endereco_cliente.clientecpfclnt IS 'chave
estrangeira que identifica a qual cliente está vinculado o
endereco';

```

```

CREATE TABLE funcionario (
    codfun SERIAL NOT NULL
    ,
    nomfun varchar(40) NOT NULL
    ,
    logfun varchar(20) NOT NULL UNIQUE
    ,
    senfun varchar(32) NOT NULL,
    PRIMARY KEY (codfun));
COMMENT ON TABLE funcionario IS 'detalhamento do
funcionario/administrador do sistema';
COMMENT ON COLUMN funcionario.codfun IS 'código do
funcionario, gerado automaticamente de forma sequencial';
COMMENT ON COLUMN funcionario.nomfun IS 'nome do funcionario';
COMMENT ON COLUMN funcionario.logfun IS 'login do
funcionario';
COMMENT ON COLUMN funcionario.senfun IS 'senha do
funcionario';
CREATE TABLE pedido (
    codped SERIAL NOT NULL,
    clientecpfclnt numeric(11, 0) NOT NULL,
    PRIMARY KEY (codped));
COMMENT ON TABLE pedido IS 'tabela que contem o codigo de
identificação do pedido e a qual cliente ele está vinculado.
Lembrando que nem todo pedido é uma venda.';
COMMENT ON COLUMN pedido.codped IS 'codigo do pedido';
COMMENT ON COLUMN pedido.clientecpfclnt IS 'chave estrangeira
a qual vincula o cpf ao pedido';
CREATE TABLE pedido_produto (
    itempedidoprod SERIAL NOT NULL,
    produtocodprod int4 NOT NULL,
    pedidocodped int4 NOT NULL,
    qntped int4 NOT NULL,
    PRIMARY KEY (itempedidoprod));
COMMENT ON TABLE pedido_produto IS 'tabela que relaciona os
produtos selecionados ao pedido. pode existir mais de um
produto no mesmo pedido';
COMMENT ON COLUMN pedido_produto.itempedidoprod IS 'item do
produto no pedido (ex: 1,2,3...)';
COMMENT ON COLUMN pedido_produto.produtocodprod IS 'chave
estrangeira que identifica o produto';
COMMENT ON COLUMN pedido_produto.pedidocodped IS 'chave
estrangeira que identifca o pedido';
COMMENT ON COLUMN pedido_produto.qntped IS 'quantidade de
itens pedidos do mesmo produto';
CREATE TABLE produto (
    codprod SERIAL NOT NULL
    ,
    desprod varchar(140),
    nomprod varchar(40) NOT NULL
    ,
    valprod numeric(10, 2) NOT NULL,
    estprod int4 NOT NULL,
    imgprod varchar(80),
    PRIMARY KEY (codprod));

```

```

COMMENT ON TABLE produto IS 'informacoes do produto';
COMMENT ON COLUMN produto.codprod IS 'codigo do produto gerado
de forma sequencial';
COMMENT ON COLUMN produto.desprod IS 'descricao do produto';
COMMENT ON COLUMN produto.nomprod IS 'nome do produto';
COMMENT ON COLUMN produto.valprod IS 'valor do produto(R$)';
COMMENT ON COLUMN produto.estprod IS 'quantidade em estoque do
produto';
COMMENT ON COLUMN produto.imgprod IS 'nome de imagem
cadastrada na pasta imgproduto';
CREATE TABLE telefone_cliente (
    codtelclnt      SERIAL NOT NULL,
    dddtelclnt      numeric(2, 0) NOT NULL,
    numtelclnt      numeric(9, 0) NOT NULL,
    clientecpfclnt  numeric(11, 0) NOT NULL,
    PRIMARY KEY (codtelclnt));
COMMENT ON TABLE telefone_cliente IS 'telefones do cliente.
pode existir mais de um registro';
COMMENT ON COLUMN telefone_cliente.codtelclnt IS 'codigo do
telefone do cliente. gerado automaticamente de forma
sequencial';
COMMENT ON COLUMN telefone_cliente.dddtelclnt IS 'DDD do
telefone do cliente.';
COMMENT ON COLUMN telefone_cliente.numtelclnt IS 'numero do
telefone cadastrado para o cliente';
COMMENT ON COLUMN telefone_cliente.clientecpfclnt IS 'chave
estrangeira que identifica a qual cliente está vinculado o
telefone';
CREATE TABLE venda (
    codvenda        SERIAL NOT NULL,
    pedidocodped    int4 NOT NULL,
    valvenda        numeric(6, 2) NOT NULL,
    stsvenda        varchar(140) NOT NULL,
    datvenda        date NOT NULL,
    horvenda        time NOT NULL,
    PRIMARY KEY (codvenda));
COMMENT ON TABLE venda IS 'detalhamento da venda (pedido
efetuado)';
COMMENT ON COLUMN venda.codvenda IS 'codigo da venda';
COMMENT ON COLUMN venda.pedidocodped IS 'codigo do pedido ao
qual a venda está relacionada';
COMMENT ON COLUMN venda.valvenda IS 'valor total da venda';
COMMENT ON COLUMN venda.stsvenda IS 'status da venda';
COMMENT ON COLUMN venda.datvenda IS 'data venda';
COMMENT ON COLUMN venda.horvenda IS 'horario da venda';
ALTER TABLE pedido_produto ADD CONSTRAINT FKpedido_pro231850
FOREIGN KEY (produtocodprod) REFERENCES produto (codprod);
ALTER TABLE pedido_produto ADD CONSTRAINT FKpedido_pro330558
FOREIGN KEY (pedidocodped) REFERENCES pedido (codped);

```

```
ALTER TABLE venda ADD CONSTRAINT FKvenda21377 FOREIGN KEY  
(pedidocodped) REFERENCES pedido (codped);  
ALTER TABLE pedido ADD CONSTRAINT FKpedido77747 FOREIGN KEY  
(clientecpfclnt) REFERENCES cliente (cpfclnt);  
ALTER TABLE telefone_cliente ADD CONSTRAINT FKtelefone_c632501  
FOREIGN KEY (clientecpfclnt) REFERENCES cliente (cpfclnt);  
ALTER TABLE endereco_cliente ADD CONSTRAINT FKendereco_c624907  
FOREIGN KEY (clientecpfclnt) REFERENCES cliente (cpfclnt);
```

ANEXO C – SCRIPT SQL PARA APAGAR TABELAS

```
ALTER TABLE pedido_produto DROP CONSTRAINT FKpedido_pro231850;
ALTER TABLE pedido_produto DROP CONSTRAINT FKpedido_pro330558;
ALTER TABLE venda DROP CONSTRAINT FKvenda21377;
ALTER TABLE pedido DROP CONSTRAINT FKpedido77747;
ALTER TABLE telefone_cliente DROP CONSTRAINT
FKtelefone_c632501;
ALTER TABLE endereco_cliente DROP CONSTRAINT
FKendereco_c624907;
DROP TABLE IF EXISTS cliente CASCADE;
DROP TABLE IF EXISTS endereco_cliente CASCADE;
DROP TABLE IF EXISTS funcionario CASCADE;
DROP TABLE IF EXISTS pedido CASCADE;
DROP TABLE IF EXISTS pedido_produto CASCADE;
DROP TABLE IF EXISTS produto CASCADE;
DROP TABLE IF EXISTS telefone_cliente CASCADE;
DROP TABLE IF EXISTS venda CASCADE;
```