**GABRIEL DOMINGOS**

**MAICON SANTOS DA SILVA**

**MATEUS CLEMER QUINTINO**

ESTRUTURA DE BANCO DE DADOS PARA E-COMMERCE

**BLUMENAU**

**2017**

###### SUMÁRIO

Sumário

[1 INTRODUÇÃO 1](#_Toc460259029)

[2 DESENVOLVIMENTO 2](#_Toc460259030)

[2.1 Requisitos do Sistema 2](#_Toc460259031)

[2.2 Modelo Entidade Relacionamento 2](#_Toc460259032)

[2.3 Implementação do Sistema 3](#_Toc460259033)

[3 CONSIDERAÇÕES FINAIS 7](#_Toc460259034)

[REFERÊNCIAS BIBLIOGRÁFICAS 8](#_Toc460259035)

# 

# 1 INTRODUÇÃO

O problema inicial apresentado é controlar as vendas de produtos e supervisionar o estoque dos mesmos. Controlar os produtos que saem e que entram no estoque através de compras e vendas.

O ambiente escolhido foi WEB para que o acesso de clientes seja mais fácil e para que as vendas sejam registradas automaticamente. Usuários novos podem se cadastras sozinhos, realizar vendas e consultar os produtos. E para gerenciar o estoque outras de produtos e cadastro de transportadores existe uma área separada de administradores, no estilo intranet.

# 2 DESENVOLVIMENTO

Neste capítulo são apresentados os requisitos do sistema, o Modelo Entidade Relacionamento, os trechos de códigos no contexto de banco de dados, seguido de imagens ilustrando o resultado das operações.

# 2.1 Requisitos do Sistema

O sistema SharpStore é um sistema web de vendas. O sistema é acessado tanto pelo cliente quanto pelos administradores e cada usuário tem suas funções limitadas por essas duas categorias. Cada usuário possui um login e senha próprio.

O sistema é composto por:

* Cadastros: para inclusão, edição produtos, usuário, transportadora e vendas.
* Consultas: as consultas de produtos, clientes e vendas podem ser feitas pelos administradores numa área separada, já para clientes a consulta do produto é disponível na hora de fazer a venda.

Como requisitos principais, o sistema deve permitir que os clientes criem seus próprios perfis (usuário cliente), permitindo, ainda, editarem o mesmo. E as funções de consultar e deletar usuários, ou criar um usuário administrador, é permitida somente para administradores.

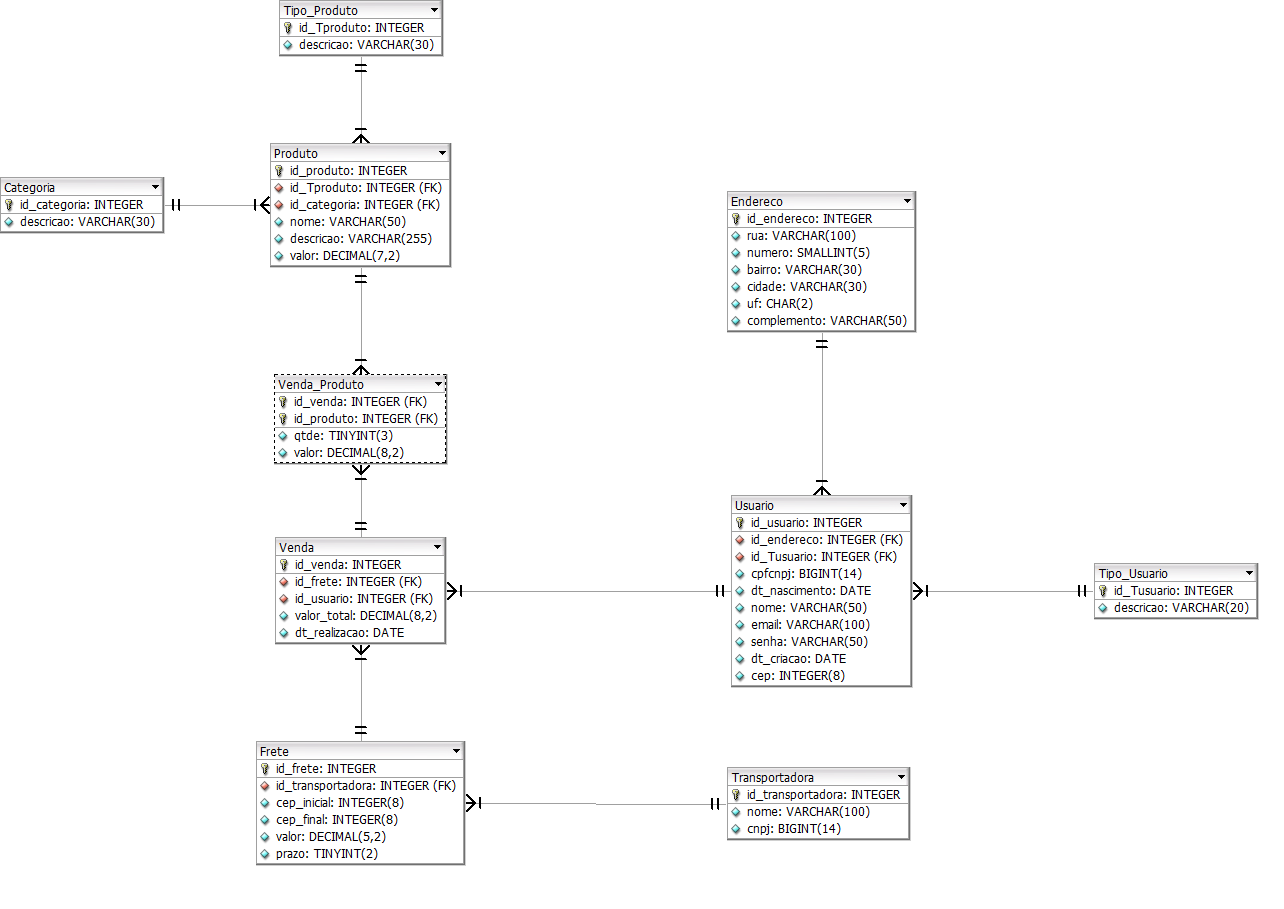
O sistema deve permitir que usuário já cadastrados consultem os produtos previamente cadastros e adicionem no carrinho de compra, ao finalizar a compra deve ser possível seleciona o frete.

O inclusão, edição e remoção dos produtos e das transportadoras é função exclusiva dos administradores. O cadastro de produto é composto por seu nome, descrição, valor e tipo. Já as transportadoras por seu nome e CNPJ.

# 2.2 Modelo Entidade Relacionamento

Com base nas informações coletadas e no levantamento de requisitos funcionais foi desenvolvido o Modelo Entidade Relacionamento (MER) físico da base de dados. O MER tem como principal finalidade representar os relacionamentos de conectividade existentes para a formalização da estrutura necessária para a criação específica desta base de dados. A Figura 1 representa o diagrama resultante desenvolvido para o SGBD MySQL.

Figura 01 – Modelo Entidade Relacionamento



# 2.3 Implementação do Sistema

Ao pedir o relatório de quantidade de produtos por categoria, o sistema submete uma sentença SQL para fazer a listagem. Este trecho de código pode ser observado no quadro 1.

Quadro 01 – Código SQL de relatório de produtos por categoria

CREATE OR REPLACE

VIEW censo\_categoria\_v AS

SELECT

b.descricao,

count(\*) qtd,

max(a.valor) maior\_valor

FROM

Produto a,

Categoria b

WHERE

a.id\_categoria = b.id\_categoria

GROUP BY

b.descricao

HAVING

count(\*) > 1

;

Ao pedir o relatório de quantidade de produtos por tipo, o sistema submete uma sentença SQL para fazer a listagem. Este trecho de código pode ser observado no quadro 2.

Quadro 02 – Código SQL de relatório de produtos por tipo

CREATE OR REPLACE

VIEW censo\_tipo\_prod\_v AS

SELECT

b.descricao,

count(\*) qtd,

max(a.valor) maior\_valor

FROM

Produto a,

Tipo\_Produto b

WHERE

a.id\_categoria = b.id\_categoria

GROUP BY

b.descricao

HAVING

count(\*) > 1

;

Ao pedir a relação do valor total de um produto específico, o sistema submete uma sentença SQL para fazer a relação. Este trecho de código pode ser observado no quadro 3.

Quadro 03 – Código SQL da relação do valor total do produto

CREATE OR REPLACE

VIEW rel\_venda\_v AS

SELECT

c.nome,

a.valor\_total,

b.valor valor\_unitario,

d.valor valor frete

FROM

Venda a,

Venda\_Produto b,

Usuario c,

Frete d

WHERE

a.id\_venda = b.id\_venda AND

a.id\_usuario = v.id\_usuario AND

a.id\_frete = d.id\_frete

;

SELECT \*

FROM

rel\_venda\_v a

WHERE

a.id\_venda = :id\_venda

ORDER BY

a.nome

;

Ao pedir o relatório de usuários clientes criados em um período, o sistema submete uma sentença SQL para fazer a listagem. Este trecho de código pode ser observado no quadro 4.

Quadro 04 – Código SQL de relatório de usuários clientes

CREATE OR REPLACE

VIEW cliente\_v AS

SELECT

a.nome,

a.email

FROM

Usuario a

WHERE

a.id\_Tusuario = 2

;

SELECT \*

FROM

cliente\_v a

WHERE

a.dt\_criacao

BETWEEN

:dt\_inicio AND

:dt\_fim

ORDER BY

a.nome

;

Antes de finalizar a venda, o sistema submete uma sentença SQL para mostrar ao usuário as opções de frete disponíveis para o CEP do mesmo. Este trecho de código pode ser observado no quadro 5.

Quadro 05 – Código SQL da listagem de frete disponível por CEP

CREATE OR REPLACE

VIEW frete\_transp\_v AS

SELECT

a.nome,

b.cep\_inicial,

b.cep\_final,

b.valor

FROM

Transportadora a,

Frete b

WHERE

a.id\_transportadora = b.id\_transportadora

GROUP BY

a.nome,

b.cep\_inicial,

b.cep\_final,

b.valor

;

SELECT \*

FROM

Frete\_transp\_v a

WHERE

a.cep\_inicial <= :cep AND

a.cep\_final >= :cep

;

Ao administrador pedir o relatório de vendas nos últimos 30 dias, o sistema submete uma sentença SQL para fazer a listagem. Este trecho de código pode ser observado no quadro 6.

Quadro 06 – Código SQL de relatório de vendas

CREATE OR REPLACE

VIEW ultimas\_vendas \_v AS

SELECT

b.nome,

a.id\_venda cod\_venda,

a.valor\_total

FROM

Venda a,

Usuario b

WHERE

a.id\_usuario = b.id\_usuario AND

a.dt\_realizacao

BETWEEN

sysdate-30 AND

sysdate

UNION

SELECT

`Valor Total`,

0,

sum(valor\_total) total\_geral

FROM

Venda a

WHERE

a.dt\_realizacao

BETWEEN

sysdate-30 AND

sysdate

;

Ao pedir o relatório de todos os produtos do banco, o sistema submete uma sentença SQL para fazer a listagem. Este trecho de código pode ser observado no quadro 7.

Quadro 07 – Código SQL de relatório de produtos

CREATE OR REPLACE

VIEW censo\_produtos\_v AS

SELECT

b.descricao ds\_categoria,

c.descricao ds\_tipo\_produto,

a.nome,

a.valor

FROM

Produto a,

Categoria b,

Tipo\_Produto c

WHERE

a.id\_categoria = b.id\_categoria AND

a.id\_Tproduto = c.id\_Tproduto

ORDER BY

1,2,3,4

;

# 3 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado o processo de desenvolvimento para um sistema web de vendas. Com o resultado obtido é possível realizar a criação de toda a estrutura necessária para comportar as necessidades levantadas nos requisitos. A partir do sistema apresentado é possível gerenciar todas as vendas feitas pela loja e consultar os estoques e informações que serão pelos clientes para realizarem compras.

A criação do software sugere facilitas a venda dos produtos, fazendo com que os clientes consigam comprar de forma rápida e de sua casa. As tecnologias usadas são html, css e javascript para front-end e para backend um servidor python com banco de dados MySQL.

# REFERÊNCIAS BIBLIOGRÁFICAS

VALDAMERI, A. R. **Materiais sobre Banco de Dados.**. Disponível em: <<http://ava.furb.br/ava2/FURB/inicial/>>. Acesso em: 01 mar. 2017.