

## Introdução:

**Loja virtual** designa uma página na Internet com um software de gerenciamento de pedidos (carrinho de compras ou cesto de compras) na qual empresas oferecem e vendem seus produtos. Os clientes acessam o site, escolhem os produtos para aquisição e recebem estes produtos em casa. Para ser caracterizada como loja virtual, o site deve ter todo o processo feito pela internet, desde a escolha dos produtos até o pagamento.

Fonte: [http://pt.wikipedia.org/wiki/Loja\\_virtual](http://pt.wikipedia.org/wiki/Loja_virtual)

Esse trabalho tem como objetivo construir uma loja virtual que atenda a definição acima. Para tal foram propostas as seguintes funcionalidades:

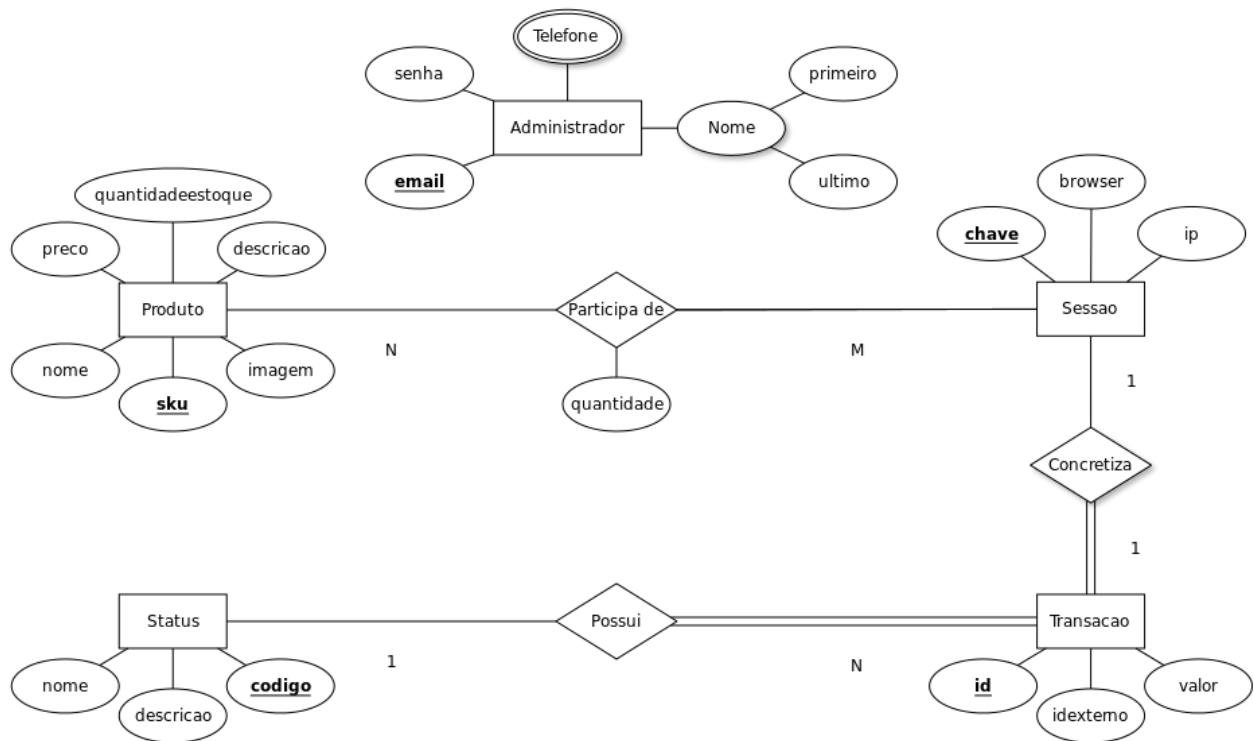
1. Ambiente Administrativo: inserção de produtos,
2. Ambiente Público: catálogo de produtos,
3. Controle de Vendas: usando sessão, sem autenticação,
4. Transação de Pagamento: implementada por terceiros.

Além da seguinte especificação:

Construir uma loja virtual em dois ambientes: o primeiro, no qual o **administrador** insere um **produto** e o segundo, em que o cliente pode buscar por ele. O cliente não precisa se autenticar, uma vez que os produtos escolhidos *participarão* de uma **sessão**. Ao término da escolha, o cliente será encaminhado (junto com os dados da compra) a um serviço de pagamento online, que *concretiza* a **sessão** em uma **transação** que *possui* um **status**.

## Características do Banco:

### Diagrama ER:



## **Modelo Relacional:**

O mapeamento do Diagrama ER acima resultou no seguinte Modelo Relacional

```
administrador(email, senha, nome, sobrenome)

telefoneadmin(email, telefone)
    email referencia administrador.email

imagens(imagem, dadosbase64)

produto(sku, nome, descricao, preco, imagem)
    imagem referencia imagens.imagem

sessao(chave, ip, browser)

carrinho(chave, sku, quantidade)
    chave referencia sessao.chave
    sku referencia produto.sku

status(codigo, nome, descricao)

transacao(id, idexterno, valor, codigostatus, chavesessao)
    codigostatus referencia status.codigo
    chavesessao referencia sessao.chave
```

Para os relacionamentos existentes dois deles (Possui e Concretiza) foram mapeados com adição de coluna na tabela *transação*, uma vez que a participação desta entidade é total e a cardinalidade das outras entidades para com ela é *única*. Já para o terceiro (Participa De) foi criada uma tabela própria (a tabela carrinho), por conta da cardinalidade M:N e também pelo atributo quantidade que não poderia pertencer a nenhuma das duas tabelas envolvidas.

As imagens são armazenadas no próprio banco de dados codificadas no formato base64. Elas são decodificadas quando necessário para exibição.

## Consultas Realizadas:

### *Apenas junções e projeções:*

1) Todos dados de produtos em estoque:

```
SELECT *  
FROM produto  
WHERE quantidadeestoque > 0  
ORDER BY nome
```

2) Nome e sobrenome do administrador com o e-mail dado:

```
SELECT nome, sobrenome  
FROM administrador  
WHERE email='dadoEmail'
```

### *Uma junção entre duas relações:*

3) Selecione os identificadores de todas transações no Google Chrome que estão aguardando pagamento:

```
SELECT *  
FROM sessao INNER JOIN transacao ON sessao = chavesessao  
WHERE codigostatus = 4 AND browser LIKE '%Google Chrome%'
```

4) Selecione os endereços IP e o valor de todas as transações com valores superiores a R\$1000,00:

```
SELECT ip, valor  
FROM sessao INNER JOIN transacao ON sessao = chavesessao  
WHERE valor > 1000.00
```

5) O Nome e os telefones de todos administradores com senhas contendo menos de 8 caracteres:

```
SELECT nome, telefone  
FROM administrador NATURAL JOIN telefoneadmin  
WHERE length(senha) < 8
```

### *Duas ou mais junções:*

6) Selecione todos os nomes dos produtos que foram adicionados a carrinhos diversos mas que a sessão não foi concretizada:

```
SELECT nome  
FROM produto NATURAL JOIN carrinho NATURAL JOIN sessao  
WHERE chave NOT IN (  
    SELECT chave  
    FROM sessao INNER JOIN transacao ON sessao =  
chavesessao)
```

7) Todos os identificadores de transações envolvendo SSDs que foram canceladas ou que não foram pagas:

```
SELECT id  
FROM (produto NATURAL JOIN carrinho NATURAL JOIN sessao)  
      INNER JOIN transacao ON sessao = chavesessao  
WHERE codigostatus = 2 OR codigostatus = 3 AND produto.sku = `SSD`
```

8) Os IPs de todas transações com 100 ou mais unidades de qualquer produto que estão aguardando pagamento:

```
SELECT ip  
FROM (carrinho NATURAL JOIN sessao) INNER JOIN transacao ON sessao  
      = chavesessao  
WHERE quantidade >= 100 AND codigostatus = 4
```

### ***Agregação:***

9) Contar quantas transações foram concluídas em cada IP e retornar o valor máximo e a média dos valores das transações destes IPs:

```
SELECT ip, COUNT(*), MAX(valor), AVG(valor)  
FROM sessao INNER JOIN transacao ON sessao = chavesessao  
GROUP BY ip
```

10) Achar qual é o nome do produto mais vendido e a quantidade vendida:

```
SELECT nome, MAX(qtd)  
FROM (SELECT sku, nome, SUM(quantidade) as qtd  
      FROM (produto NATURAL JOIN carrinho NATURAL JOIN sessao)  
          INNER JOIN transacao ON sessao = chavesessao  
      WHERE codigostatus >= 5  
      GROUP BY sku)
```

## **Avaliação:**

Manassés avalia Marzo:

9,137 em 10. Pelo descuido com o prazo do relatório final.

Marzo avalia Manassés

$9 + 1/\pi$  em 10 devido à perda de prazo.