

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

BANCO DE DADOS

CIÊNCIA DA COMPUTAÇÃO

ANDRÉ FONSECA DE PAIVA

ANGELICA DOS SANTOS

LUCAS OMAR ANDRADE LEAL

RHIAN LUIS GARCIA MORAES

YURY REGIS NEIVA PEREIRA

PRIMEIRO TRABALHO

ETAPA FINAL

POÇOS DE CALDAS - MG

2021

Trabalho de Banco de Dados

1. Visão do Produto:

Site de classificados de eletrônicos o qual o usuário pode vender demasiados produtos ou adquiri-los. Deverá ter um perfil de usuário, onde o mesmo pode ser comprador e/ou vendedor. Um usuário pode vender diversos produtos. Terá um cadastro de produtos, com valores, descrição do produto, data de inclusão, data de venda, status de vendido, disponível, ou indisponível.

1.1 Descrição dos Usuários:

Papel	Descrição do Papel do Usuário
Usuário Geral	Cadastrar seus dados, ser um comprador ou um vendedor.
Comprador	Adicionar compras ao carrinho temporário, pagar compras do pedido.
Vendedor	Cadastrar produtos.

2. Requisitos Funcionais

Fornecer uma lista de enumeração das funções que o sistema deverá fornecer aos usuários. Identificar tanto requisitos de entrada, de processamento e de saída. Escreva pelo menos 20 requisitos no seguinte formato.

Código	Descrição	Obrigatório (S/N)
RF01	Só deve autorizar uma compra se o usuário tiver o endereço cadastrado.	S
RF02	O sistema deve obrigar o usuário a cadastrar seu CPF.	S
RF03	Permitir que o usuário veja uma lista dos produtos que são vendidos no site.	S
RF04	Permitir que um usuário comprador possa adicionar vários produtos ao carrinho temporário, validando se a quantidade do produto em estoque bate a quantidade que será adicionada no carrinho temporário.	S
RF05	Fornecer suporte para informar acerca do status de um pedido, e informar se a compra foi bem-sucedida.	N
RF06	O sistema deve permitir que a compra seja cancelada.	N
RF07	Permitir que um usuário possa remover produtos do pedido.	N
RF08	Permitir que o usuário pesquise por um determinado	N

	produto.	
RF10	Permitir ver os produtos de um determinado vendedor.	N
RF11	Permitir que o usuário realize pesquisas mesmo sem estar logado.	N
RF13	Com a confirmação do pagamento de qualquer compra, o sistema deve adicionar o lucro desta compra ao saldo do vendedor.	S
RF14	Com a confirmação do pagamento de qualquer compra, o sistema deve debitar o valor do usuário.	S
RF15	Possibilitar a listagem dos produtos por categoria.	N
RF16	Permitir que vendedor informe se é representante de uma loja oficial ou não	N
RF17	Apresentar quantos produtos estão próximos ao comprador (através do CEP).	N
RF18	Apresentar a quantidade de produtos disponíveis em uma determinada listagem.	S
RF19	Deve remover anúncios expirados.	N
RF20	Pode possibilitar a consulta de histórico através da tabela de pagamentos.	N
RF21	Permitir a apresentação de relatórios dos produtos mais vendidos	N
RF22	Permitir a apresentação de relatórios de qual fornecedor mais vendeu e menos vendeu	N
RF23	Permitir a ordenação dos produtos por preço	N

2. Identifique as cardinalidades dos relacionamentos.

Entidade 1	Relacionamento	Entidade 2	Cardinalidade
Usuário	vende	Produto	1:n
Pedido	contém	Produto	n:n
Usuário	possui	Pedido	1:n
Usuário	efetua	Pagamento	1:n
Pagamento	se trata de	Pedido	1:1
Pagamento	possui	Status	n:1
Produto	corresponde a	Categoria	n:n

3. Projeto Conceitual do Banco de Dados

3.1. Identifique as entidades que o banco de dados precisar se lembrar.

- Produto
- Usuário > (Comprador, Vendedor)
- Pedido
- Pagamento
- Categorias

USUÁRIO

O usuário é quem acessa o sistema, ele pode ter tanto a função de comprador quanto de vendedor, simultaneamente.

DESCRIÇÃO DOS ATRIBUTOS

Nome	Descrição	Domínio	Tamanho	ID (S/N)	Obrigatório (S/N)
Nome	Nome do usuário	VARCHAR	45	S	S
CPF	Código de pessoa física	CHAR	11	N	S
Endereço	Residência do usuário	VARCHAR	45	N	S
CEP	Código de endereço Postal	CHAR	8	N	S
E-mail	Endereço eletrônico do usuário	VARCHAR	45	N	S
Telefone	Telefone de contato	CHAR	11	N	S
Saldo	Valor em BRL do saldo em conta do usuário	DECIMAL	8,2	N	S
Flag Vendedor	Classificação do usuário	BOOL	-	N	S
Flag Comprador	Classificação do usuário	BOOL	-	N	S
Flag Loja Oficial	Classificação do usuário	BOOL	-	N	N

PRODUTO

Os produtos que serão vendidos ou comprados pelos usuários. Se conecta com o pedido para que seja apresentado no pedido. Também se conecta ao pagamento para que se tenha um registro do que foi comprado.

DESCRIÇÃO DOS ATRIBUTOS

Nome	Descrição	Domínio	Tamanho	ID (S/N)	Obrigatório (S/N)
código	código do	TINYINT	2b	S	S

	produto				
título	título do produto	CHAR	30	N	S
descrição	descrição do produto	CHAR	2000	N	S
preço	valor do produto	DECIMAL	8,2	N	S
quantidade	quantidade de produtos	INT	8	N	S
categoria	tipo do produto	CHAR	20	N	S
data de expiração	data final	DATE	-	N	S

PEDIDO

Onde serão incluídos os produtos para efetivação da compra. Quando um usuário navega no site, ele adiciona os produtos a um carrinho temporário, que só existirá no frontend. Para existir um pedido no banco de dados, ele deverá clicar em Comprar e realmente efetuar o pagamento.

DESCRIÇÃO DOS ATRIBUTOS

Nenhum atributo próprio.

Nome	Descrição	Domínio	Tamanho	ID (S/N)	Obrigatório (S/N)
código do pedido	Atributo identificador	INT	2B	S	S
Qtd_produto	quantidade de produtos no pedido	INT	1	N	S

PAGAMENTO

O pagamento é o registro que representa a compra do usuário.

DESCRIÇÃO DOS ATRIBUTOS

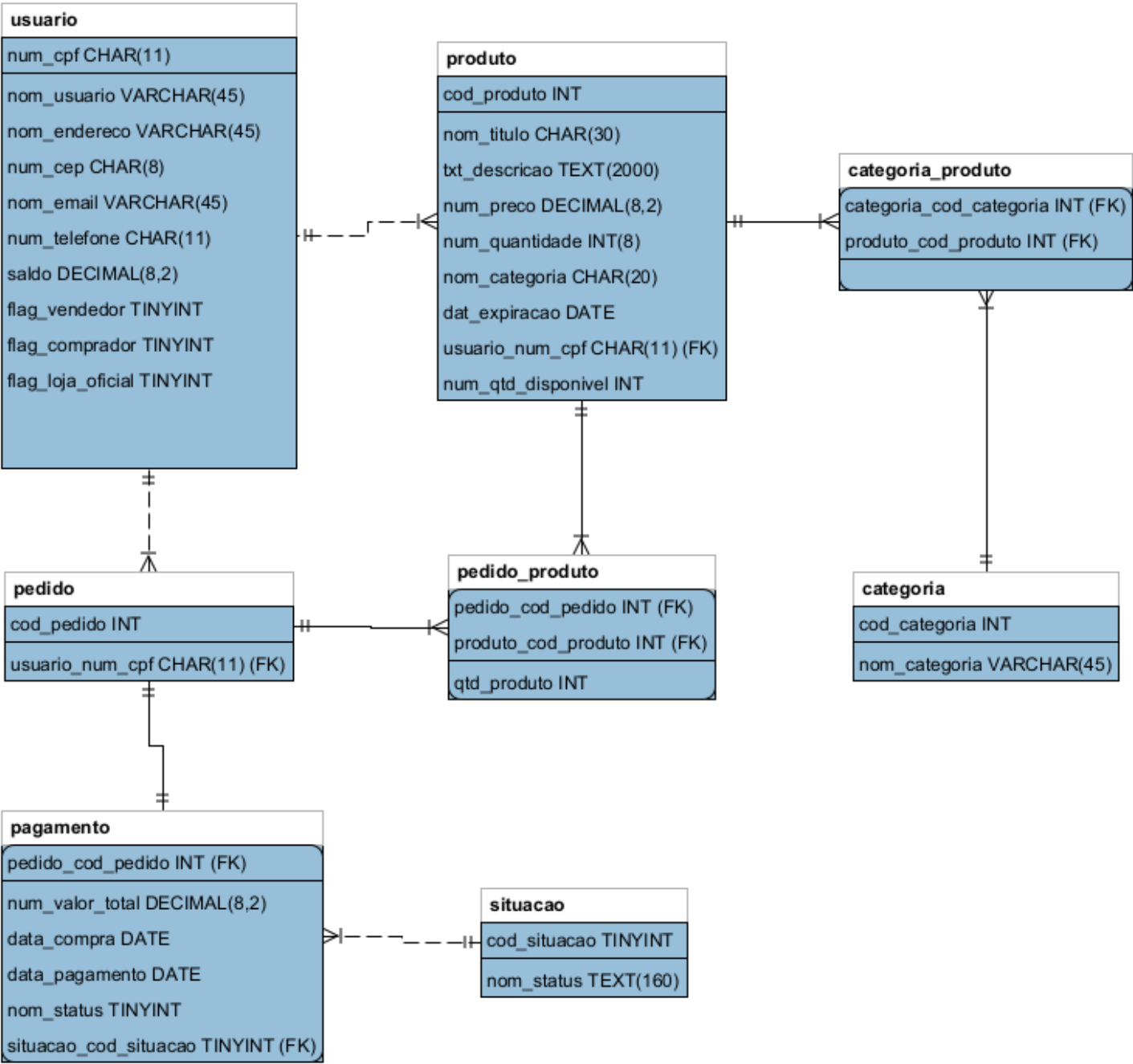
Nome	Descrição	Domínio	Tamanho	ID (S/N)	Obrigatório (S/N)
código do pedido	código do pedido referente	INT	2B	S	S
valor total	valor total do pagamento	DECIMAL	8,2	N	S
data da compra	data do pedido de compra	DATE	-	N	S
data do pagamento	data que o pagamento foi realizado	DATE	-	N	N
status	caso a compra ocorra com sucesso ou não	BOOLEAN	-	N	S

CATEGORIA

Categoria de um determinado produto, para facilitar o crescimento do banco de dados e a adição de novas categorias.

DESCRIÇÃO DOS ATRIBUTOS

Nome	Descrição	Domínio	Tamanho	ID (S/N)	Obrigatório (S/N)
código	código identificador	INT	2 bytes	S	S
nome	nome da categoria	TEXT	40 caracteres	N	S



4.2 Registro de decisões

Mescla de Requisitos

A princípio foi apresentada a ideia de informar, em um requisito isolado, se a compra foi bem sucedida. A fim de deixar a produção mais direta e enxuta, foi adicionado tal finalidade no requisito de compra.

Verificação da quantidade de produtos integrada à requisição de possibilidade de adicionar ao carrinho.

Substituição de 'atributos'

Alteração de booleans com prefixo 'is_%' para 'flag_%' a fim de atender as boas práticas da programação.

Ex: is_vendedor → flag_vendedor
 is_comprador → flag_comprador
 is_loja_oficial → flag_loja_oficial

Adaptação do carrinho. Essa propriedade torna-se a entidade **pedido** a fim de otimização e pragmatização do código.

Alteração do tipo de CHAR para TEXT para deixar mais versátil e ideal para o uso.
txt_descricao CHAR() → txt_descricao TEXT()

Alteração de TEXT para VARCHAR para ficar ideal para este uso

Ex: nom_categoria TEXT(45) → nom_categoria VARCHAR(45)

Alteração para ficar mais próximo ao padrão utilizado

Ex: PRIMARY KEY(cod_categoria)

Alteração do nome de 'status' para 'situacao' pois status é uma palavra reservada

Ex:

```
CREATE TABLE situacao (  
    cod_situacao TINYINT,
```

Alteração para tipo VARCHAR para melhor atender às demandas

Ex: nom_status VARCHAR(90)

Criação da chave artificial cod_produto.

Avaliação de propriedades essenciais

Remoção das entidades **busca_efetivada** e **carrinho**. A entidade de busca não se faz essencial ao **bd**.

Entidade carrinho transformada em pedido.

5. Projeto físico e implementação

5.1 Código para implementação do banco de dados

```
DROP DATABASE IF EXISTS loja;
```

```
CREATE DATABASE loja;
```

```
USE loja;
```

```
CREATE TABLE usuario (  
    num_cpf          CHAR(11) NOT NULL,  
    nom_usuario      VARCHAR(45),  
    nom_endereco     VARCHAR(45),  
    num_cep          CHAR(8),  
    nom_email        VARCHAR(45),  
    num_telefone     CHAR(11),  
    saldo            DECIMAL(8,2),  
    flag_vendedor    TINYINT,  
    flag_comprador   TINYINT,  
    flag_loja_oficial TINYINT,
```

```
    CONSTRAINT pk_usuario  
        PRIMARY KEY(num_cpf)
```

```
) ENGINE = "InnoDB";
```

```
CREATE TABLE produto (  
    cod_produto      INT, -- Chave artificial  
    nom_titulo       CHAR(30),  
    txt_descricao    TEXT(2000), -- Alteração do tipo de CHAR para TEXT para  
deixar mais versátil e ideal para o uso  
    num_preco        DECIMAL(8,2),  
    num_quantidade   INT(8),  
    nom_categoria    CHAR(20),  
    dat_expiracao    DATE,  
    num_qtd_disponivel INT,  
    usuario_num_cpf  CHAR(11),
```

```
    CONSTRAINT pk_produto  
        PRIMARY KEY(cod_produto),
```

```
    CONSTRAINT fk_produto_usuario  
        FOREIGN KEY(usuario_num_cpf)  
        REFERENCES usuario(num_cpf)
```

```
) ENGINE = "InnoDB";
```

```
CREATE TABLE categoria (  
    cod_categoria INT,  
    nom_categoria VARCHAR(45), -- Alteração de TEXT para VARCHAR para ficar ideal  
    para este uso
```

```
    CONSTRAINT pk_categoria  
        PRIMARY KEY(cod_categoria) -- Alteração para ficar mais próximo ao padrão  
    utilizado  
) ENGINE = "InnoDB";
```

```
CREATE TABLE categoria_produto (  
    categoria_cod_categoria INT,  
    produto_cod_produto INT,  
  
    CONSTRAINT pk_categoria_produto  
        PRIMARY KEY(categoria_cod_categoria, produto_cod_produto),  
  
    CONSTRAINT fk_categoria_produto_categoria  
        FOREIGN KEY (categoria_cod_categoria)  
        REFERENCES categoria(cod_categoria),  
  
    CONSTRAINT fk_categoria_produto_produto  
        FOREIGN KEY (produto_cod_produto)  
        REFERENCES produto(cod_produto)  
) ENGINE = "InnoDB";
```

```
CREATE TABLE pedido (  
    cod_pedido INT,  
    usuario_num_cpf CHAR(11),  
  
    CONSTRAINT pk_pedido  
        PRIMARY KEY(cod_pedido),  
  
    CONSTRAINT fk_pedido_usuario  
        FOREIGN KEY(usuario_num_cpf)  
        REFERENCES usuario(num_cpf)  
) ENGINE = "InnoDB";
```

```
CREATE TABLE pedido_produto (  
    pedido_cod_pedido INT,  
    produto_cod_produto INT,  
    qtd_produto INT,  
  
    CONSTRAINT pk_pedido_produto  
        PRIMARY KEY(pedido_cod_pedido, produto_cod_produto),
```

```

CONSTRAINT fk_pedido_produto_pedido
    FOREIGN KEY(pedido_cod_pedido)
    REFERENCES pedido(cod_pedido),

CONSTRAINT fk_pedido_produto_produto
    FOREIGN KEY(produto_cod_produto)
    REFERENCES produto(cod_produto)
) ENGINE = "InnoDB";

-- Alteração do nome de 'status' para 'situacao' pois status é uma palavra
reservada
CREATE TABLE situacao (
    cod_situacao TINYINT,
    nom_status VARCHAR(90), -- Alteração para tipo VARCHAR para melhor atender às
demandas

    CONSTRAINT pk_situacao
        PRIMARY KEY (cod_situacao)
) ENGINE = "InnoDB";

CREATE TABLE pagamento (
    pedido_cod_pedido          INT,

    num_valor_total            DECIMAL(8,2),
    data_compra                 DATE,
    data_pagamento             DATE,
    nom_status                  TINYINT,
    situacao_cod_situacao       TINYINT,

    CONSTRAINT pk_pagamento
        PRIMARY KEY (pedido_cod_pedido),

    CONSTRAINT fk_pagamento_pedido
        FOREIGN KEY (pedido_cod_pedido)
        REFERENCES pedido(cod_pedido),

    CONSTRAINT fk_pagamento_situacao
        FOREIGN KEY (situacao_cod_situacao)
        REFERENCES situacao(cod_situacao)
) ENGINE = "InnoDB";

```