

INSTITUTO FEDERAL DO NORTE DE MINAS GERAIS – CAMPUS JANUÁRIA  
RODRIGO AZEVEDO MARTINS

**SISTEMA DE GERENCIAMENTO DE VENDAS**

JANUÁRIA

2013

INSTITUTO FEDERAL DO NORTE DE MINAS GERAIS – CAMPUS JANUÁRIA  
RODRIGO AZEVEDO MARTINS

## **SISTEMA DE GERENCIAMENTO DE VENDAS**

Trabalho realizado como parte de  
avaliação da disciplina de Linguagem e  
Técnicas de Programação III, ministrada  
pelo Professor Petrônio Cândido

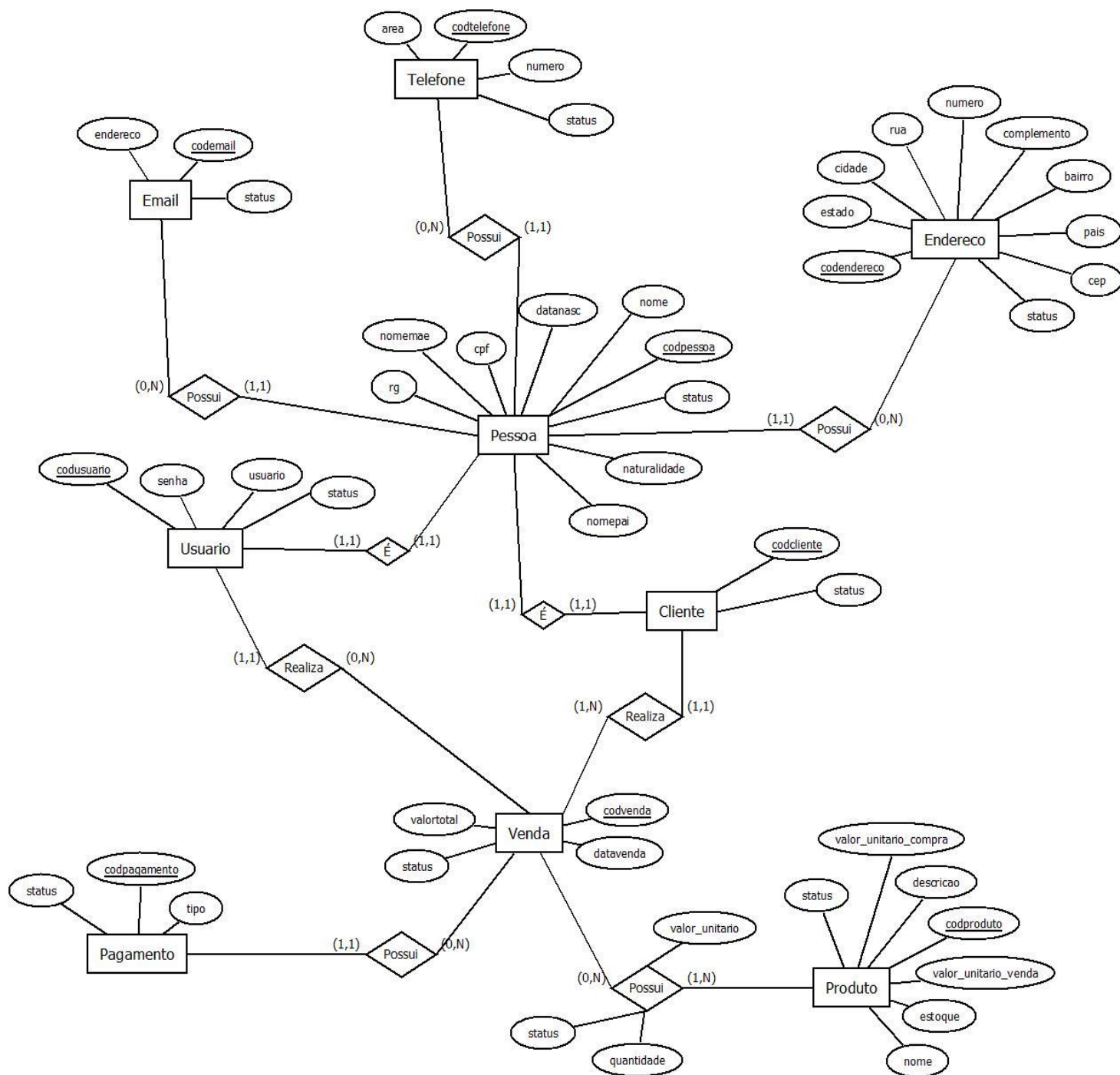
JANUÁRIA

2013

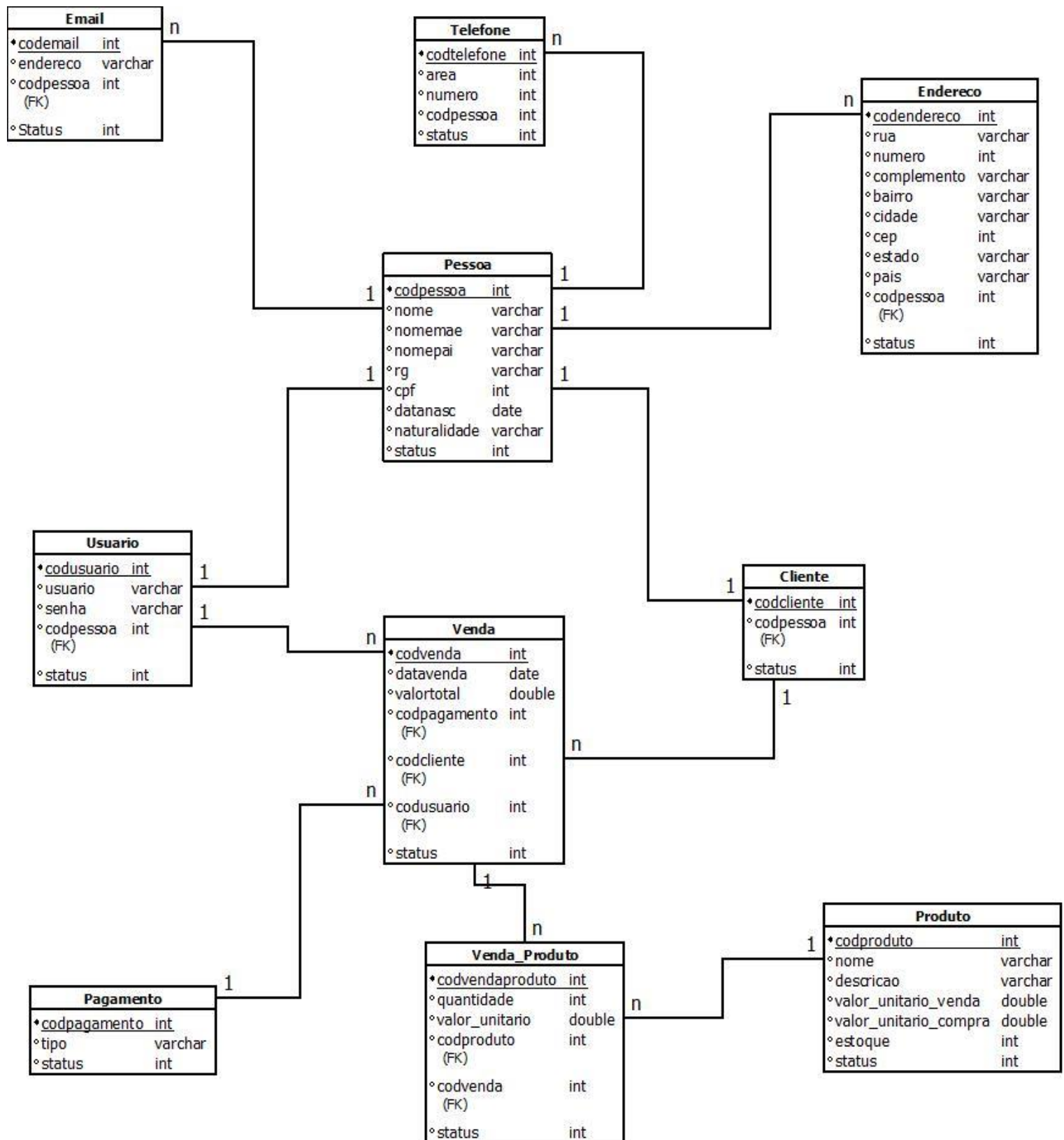
## Sumário

<b>1 MODELO ENTIDADE - RELACIONAMENTO</b> .....	4
<b>2 MODELO LÓGICO</b> .....	5
<b>3 DIAGRAMA DE CLASSES</b> .....	6
<b>4 MODELO FÍSICO</b> .....	7

# 1 MODELO ENTIDADE - RELACIONAMENTO



## 2 MODELO LÓGICO



### 3 DIAGRAMA DE CLASSES

Endereco
-codendereco: int -pessoa: Pessoa -rua: string -numero: int -complemento: string -bairro: string -cidade: string -cep: int -estado: string -pais: string
+setcodendereco(codendereco:int): void +getcodendereco(): int +setrua(rua:string): void +getrua(): string +setnumero(numero:int): void +getnumero(): int +setcomplemento(complemento:string): void +getcomplemento(): string +setbairro(bairro:string): void +getbairro(): string +setcidade(cidade:string): void +getcidade(): string +setcep(cep:int): void +getcep(): int +setestado(estado:string): void +getestado(): string +setpais(pais:string): void +getpais(): string +setpessoa(pessoa:pessoa): void +getpessoa(): pessoa

Pessoa
-codpessoa: int -nome: string -nomepai: string -nomemae: string -rg: string -cpf: int -datanasc: date -naturalidade: string -enderecos: list<Endereco> -emails: list<Email> -telefones: list<Telefone>
+setcodpessoa(codpessoa:int): void +getcodpessoa(): int +setnome(nome:string): void +getnome(): string +setnomemae(nomemae:string): void +getnomemae(): string +setnomepai(nomepai:string): void +getnomepai(): string +setrg(rg:string): void +getrg(): string +setcpf(cpf:int): void +getcpf(): int +setdatanasc(datanasc:date): void +getdatanasc(): date +setnaturalidade(naturalidade:string): void +getnaturalidade(): string +setenderecos(enderecos:list<Endereco>): void +getenderecos(): list<Endereco> +setemails(emails:list<Email>): void +getemails(): list<Email> +settelefones(telefones:list<Telefone>): void +gettelefones(): list<Telefone>

Email
-codemail: int -endereco: string -pessoa: Pessoa
+setcodemail(codemail:int): void +getcodemail(): int +setendereco(endereco:string): void +getendereco(): string +setpessoa(pessoa:pessoa): void +getpessoa(): pessoa

Telefone
-codtelefone: int -pessoa: Pessoa -area: int -numero: int
+setcodtelefone(codtelefone:int): void +getcodtelefone(): int +setarea(area:int): void +getarea(): int +setnumero(numero:int): void +getnumero(): int +setpessoa(pessoa:pessoa): void +getpessoa(): pessoa

Cliente
-codcliente: int -pessoa: Pessoa
+setcodcliente(codcliente:int): void +getcodcliente(): int +setpessoa(pessoa:pessoa): void +getpessoa(): pessoa

Venda
-codvenda: int -datavenda: date -valortotal: double -produto: Produto -cliente: Cliente -usuario: Usuario -pagamento: Pagamento -produtos: List<VendaProduto>
+setcodvenda(codvenda:int): void +getcodvenda(): int +setdatavenda(datavenda:date): void +getdatavenda(): date +setvalortotal(valortotal:double): void +getvalortotal(): double +setproduto(produto:produto): void +getproduto(): produto +setcliente(cliente:cliente): void +getcliente(): cliente +setusuario(usuario:usuario): void +getusuario(): usuario +setpagamento(pagamento:pagamento): void +getpagamento(): pagamento +setprodutos(produtos:List<VendaProduto>): void +getprodutos(): List<VendaProduto>

Usuario
-codusuario: int -usuario: string -senha: string -pessoa: Pessoa
+setcodusuario(codusuario:int): void +getcodusuario(): int +setnomeusuario(nomeusuario:string): void +getnomeusuario(): string +setsenhausuario(senhausuario:string): void +getsenhausuario(): string +setpessoa(pessoa:pessoa): void +getpessoa(): pessoa

Pagamento
-codpagamento: int -tipo: string
+setcodpagamento(codpagamento:int): void +getcodpagamento(): int +setnometipo(nometipo:string): void +getnometipo(): string

VendaProduto
-codvendaproduto: int -quantidade: int -valor_unitario: double -produto: Produto -venda: Venda
+setcodvendaproduto(codvendaproduto:int): void +getcodvendaproduto(): int +setquantidade(quantidade:int): void +getquantidade(): int +setvalorunitario(valor_unitario:double): void +getvalorunitario(): double +setproduto(produto:produto): void +getproduto(): produto +setvenda(venda:venda): void +getvenda(): venda

Produto
-codproduto: int -nome: string -descricao: string -valor_unitario_venda: double -valor_unitario_compra: double -estoque: int
+setcodproduto(codproduto:int): void +getcodproduto(): int +setnomeproduto(nomeproduto:string): void +getnomeproduto(): string +setdescricao(descricao:string): void +getdescricao(): string +setvalorunitariovenda(valorunitariovenda:double): void +getvalorunitariovenda(): double +setvalorunitariocompra(valorunitariocompra:double): void +getvalorunitariocompra(): double +setqtdestoque(qtdestoque:int): void +getqtdestoque(): int

#### 4 MODELO FÍSICO

Obs: campo status das tabelas: 0 – Inativo; 1 - Ativo

```
CREATE DATABASE sgv;
```

```
CREATE TABLE pagamento (  
    codpagamento int(11) auto_increment PRIMARY KEY,  
    tipo varchar(255),  
    status int(1)  
);
```

```
CREATE TABLE pessoa (  
    codpessoa int(11) auto_increment PRIMARY KEY,  
    nome varchar(255),  
    nomemae varchar(255),  
    nomepai varchar(255),  
    rg varchar(255),  
    cpf varchar(255),  
    datanasc date,  
    naturalidade varchar(255),  
    status int(1)  
);
```

```
CREATE TABLE telefone (  
    codtelefone int(11) auto_increment PRIMARY KEY,  
    area int(2),  
    numero int(10),
```

```
codpessoa int(11),  
FOREIGN KEY (codpessoa) REFERENCES pessoa (codpessoa),  
status int(1)  
);
```

```
CREATE TABLE email (  
codemail int(11) auto_increment PRIMARY KEY,  
endereco varchar(255),  
codpessoa int(11),  
FOREIGN KEY (codpessoa) REFERENCES pessoa (codpessoa),  
status int(1)  
);
```

```
CREATE TABLE endereco (  
codendereco int(11) auto_increment PRIMARY KEY,  
rua varchar(255),  
numero int(11),  
complemento varchar(255),  
bairro varchar(255),  
cidade varchar(255),  
cep int(8),  
estado varchar(255),  
pais varchar(255),  
codpessoa int(11),  
FOREIGN KEY (codpessoa) REFERENCES pessoa (codpessoa),  
status int(1)
```



);

```
CREATE TABLE usuario (  
    codusuario int(11) auto_increment PRIMARY KEY,  
    usuario varchar(255),  
    senha varchar(255),  
    codpessoa int(11),  
    FOREIGN KEY (codpessoa) REFERENCES pessoa (codpessoa),  
    status int(1)  
);
```

```
CREATE TABLE cliente (  
    codcliente int(11) auto_increment PRIMARY KEY,  
    codpessoa int(11),  
    FOREIGN KEY (codpessoa) REFERENCES pessoa (codpessoa),  
    status int(1)  
);
```

```
CREATE TABLE produto (  
    codproduto int(11) auto_increment PRIMARY KEY,  
    nome varchar(255),  
    descricao varchar(255),  
    valor_unitario_venda double,  
    valor_unitario_compra double,  
    estoque int(11),  
    status int(1)
```

);

CREATE TABLE venda (

codvenda int(11) auto\_increment PRIMARY KEY,

datavenda datetime,

codpagamento int(11),

codcliente int(11),

codusuario int(11),

FOREIGN KEY (codpagamento) REFERENCES pagamento (codpagamento),

FOREIGN KEY (codcliente) REFERENCES cliente (codcliente),

FOREIGN KEY (codusuario) REFERENCES usuario (codusuario),

status int(1)

);

CREATE TABLE vendaproduto (

codvendaproduto int(11) auto\_increment PRIMARY KEY,

codvenda int(11),

codproduto int(11),

quantidade int(11),

FOREIGN KEY (codvenda) REFERENCES venda (codvenda),

FOREIGN KEY (codproduto) REFERENCES produto (codproduto),

status int(1)

)

