UNiRP

CENTRO UNIVERSITÁRIO DE RIO PRETO

Projeto de Banco de Dados

Prof. Sergio Borges

MarcenariaPro
Grupo 00

Iago Lucas
Vinicius Romeiro
Matheus Miranda
Eduardo Augusto
Felipe Beltrami
Túlio Fechine

Junho de 2024

Sumário

1. Introdução

Título do Projeto: MarcenariaPro

Nome do Grupo: All Systems

Nr.	Aluno(s) envolvido(s)	E-mail(s)
01	Iago Lucas Vera Freitas	dev_iago@hotmail.com
02	Matheus Augusto Pongelupi Miranda	matheu.miranda@hotmail.com
03	Túlio Henrique Saturno Fechine	tuliofechine@gmail.com
04	Eduardo Augusto de Oliveira	eduardobuk@gmail.com
05	Vinicius Eduardo Romeiro Santos	vinicius18brasil@gmail.com
06	felipe araujo beltrami	felipe.araujo.beltrami@gmail.com

1.1 Objetivo
Construir uma aplicação desktop estilo PDV para madeireiras. Será uma aplicação white label onde cada madeireira poderá ter sua própria identidade visual.

1.2 Justificativa
O presente projeto justifica-se pela ausência de sistemas de PDVs voltados a madeireiras, fazendo com que assim surja uma necessidade no mercado e com isso nossa equipe desenvolverá tal sistema para satisfazer a necessidade dos clientes.

2. Requisitos

2.1 Descrição do Sistema

Uma madeireira faz serviços e produtos aos clientes que são previamente cadastrados. Caso o cliente não esteja cadastrado, esta atividade será realizada separadamente em uma função do sistema a partir do primeiro atendimento, contemplando para o cliente as informações de CPF, e-mail, nome, telefone, logradouro e CEP.

O cliente poderá solicitar à madeireira serviços ou produtos. Caso o cliente solicite um produto, é necessário cadastrar as informações de nome, preço, estoque, categoria, matéria prima e quantidade. Caso o cliente solicite um serviço é necessário cadastrar as informações de nome, preço e descrição.

A partir dessas informações atreladas ao atendimento, a madeireira inicia o processo de desenvolvimento do produto ou serviço solicitado, assim satisfazendo a necessidade do cliente final.

2.2 Problema a ser Resolvido

O problema a ser resolvido está relacionado à falta de sistemas de PDVs específicos a madeireiras, nosso sistema automatizará tanto para a empresa quanto para o cliente final o processo de cadastramento de solicitações de produtos e serviços, trazendo assim mais assertividade e agilidade nos pedidos e baixa perca de dados, contribuindo significantemente para o crescimento da madeireira em questão.

2.3 Não Escopo do Sistema

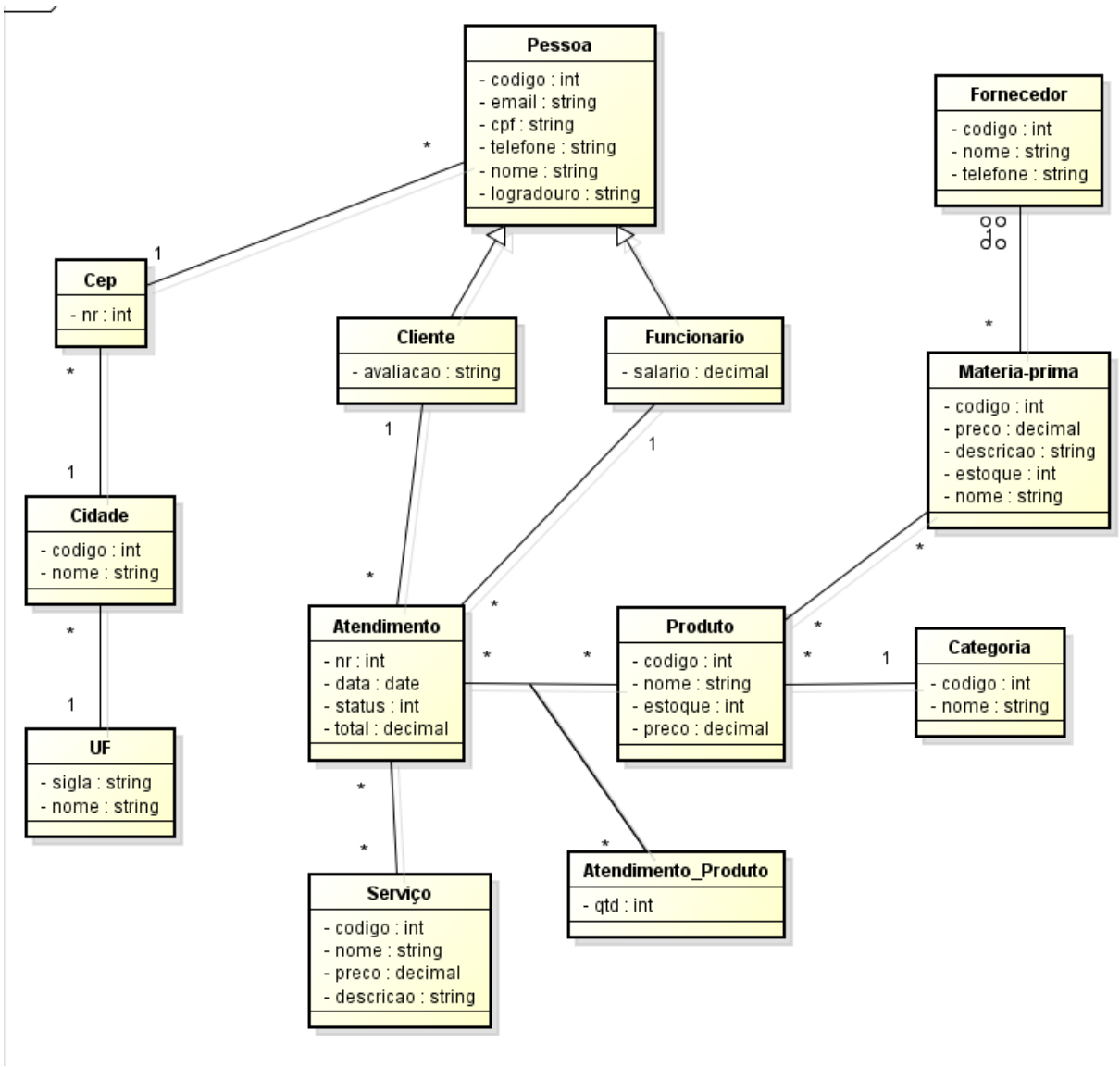
Nr.	Descrição
01	O sistema, nesse primeiro momento, não fará conexão com sistemas externos de pagamentos.

2.4 Relação dos Casos de Usos

Nr.	Descrição	Entrada	Caso de Uso	Resposta
UC 01	O atendente cadastra o cliente.	Dados do cliente.	Cadastrar Cliente	Mensagem Confirmação.
UC 02	O atendente cadastra o produto.	Dados do produto.	Cadastrar Produto	Mensagem Confirmação.
UC 03	O atendente cadastra o serviço.	Dados do serviço.	Cadastrar Serviço	Mensagem de Confirmação.
UC 04	O atendente passa o pedido à equipe de produção da madeireira.	Dados do pedido.	Envio Produção	Mensagem de Confirmação e atualização de status.
UC 06	O gerente solicita relatório de atendimentos realizados.	Período	Emitir relatório de vendas por período.	Relatório de atendimentos realizados.
UC 07	O gerente solicita relatório de faturamento.	Período	Emitir relatório de faturamento por período.	Relatório de faturamento.

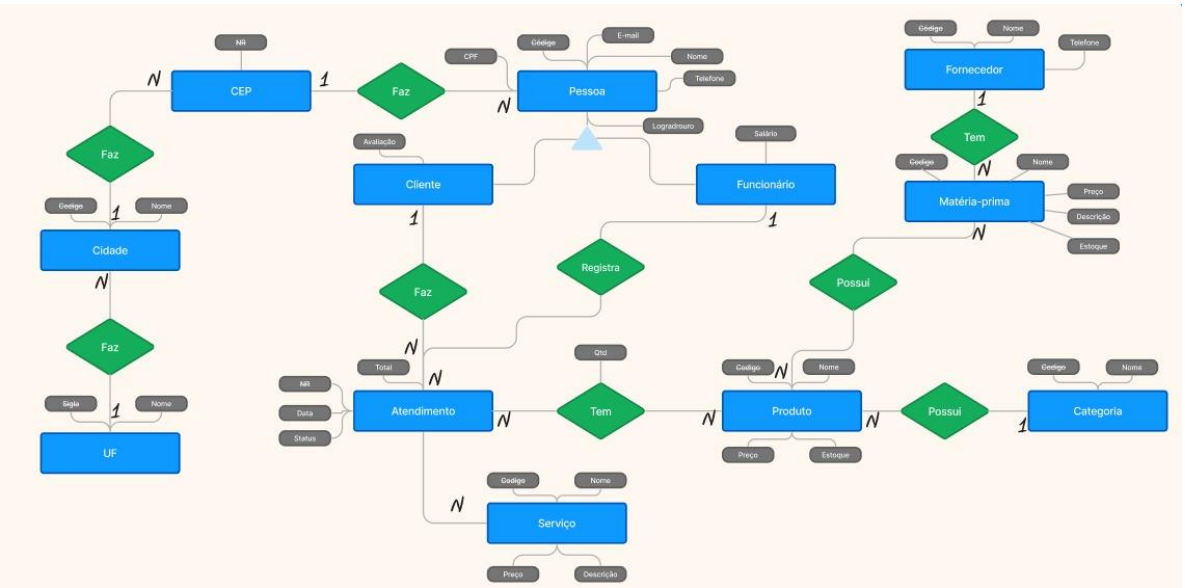
3. Modelo do Domínio

3.1 Diagrama de Classe



4. Projeto de Banco de Dados

4.1 Modelo Conceitual (Diagrama de Entidade e Relacionamento)



4.2 Modelo Lógico (Mapeamento para o Modelo Relacional)

Fornecedor (código, nome, telefone)
Matéria-prima (código, nome, preço, descrição, estoque, #fornecedor_codigo)
Categoria (código, nome)
Produto (código, nome, preço, estoque, #categoria_codigo)
UF (sigla, nome)
Cidade (código, nome, #uf_sigla)
Cep (nr, #cidade_codigo)
Pessoa (código, CPF, e-mail, nome, telefone, #cep_nr, logradouro)
Cliente (#pessoa_codigo, avaliação)
Funcionários (#pessoa_codigo, salário)
Atendimento (nr, data, status, total, #cliente_codigo, #funcionario_codigo)
Serviço (código, nome, preço, descrição, #atendimento_nr)
Produto_atendimento (#atendimento_nr, #produto_codigo)

4.3 Dicionário de Dados

Tabela	UF									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
sigla	varchar	5				no	no	no	Null	sigla
nome	varchar	30				no	no	no	Null	nome

Tabela	Cidades									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo
nome	varchar	30				no	no	no	Null	nome
sigla_uf	varchar	5				no	no	no	uf	sigla_uf

Tabela	CEP									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
Nr	varchar	10				no	no	no	Null	nr
codigo_cidade	int	4		10	0	no	(n/a)	(n/a)	cidade	codigo_cidade

Tabela	Pessoas									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo
cpf	varchar	10				no	no	no	Null	cpf
nome	varchar	30				no	no	no	Null	nome
telefone	varchar	10				yes	no	yes	Null	telefone
email	varchar	10				yes	no	yes	Null	email
logradouro	varchar	30				no	no	no	Null	logradouro
cep	varchar	10				no	no	no	Null	cep
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo

Tabela	Clientes									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
pessoa_codigo	int	4		10	0	no	(n/a)	(n/a)	pessoa	pessoa_codigo
avaliacao	varchar	50				no	no	no	Null	avaliacao

Tabela	Funcionário									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
pessoa_codigo	int	4		10	0	no	(n/a)	(n/a)	pessoa	pessoa_codigo
salario	decimal	5		7	2	no	(n/a)	(n/a)	Null	salario

Tabela	Categorias									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo
nome	varchar	50				no	no	no	Null	nome

Tabela	Fornecedores									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo
nome	varchar	50				no	no	no	Null	nome
telefone	varchar	10				no	no	no	Null	telefone

Tabela	materiasPrimas									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo
nome	varchar	30				no	no	no	Null	nome
preco	decimal	5		7	2	no	(n/a)	(n/a)	Null	preco
descricao	varchar	50				no	no	no	Null	descricao
estoque	int	4		10	0	no	(n/a)	(n/a)	Null	estoque
fornecedor	int	4		10	0	no	(n/a)	(n/a)	Null	fornecedor

Tabela	Produtos									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo
nome	varchar	50				no	no	no	Null	nome
preco	decimal	5		7	2	no	(n/a)	(n/a)	Null	preco
estoque	int	4		10	0	no	(n/a)	(n/a)	Null	estoque
categoria	int	4		10	0	no	(n/a)	(n/a)	Null	categoria

Tabela	produtosMateriasPrimas									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
produto	int	4		10	0	no	(n/a)	(n/a)	Null	produto
materiaPrima	int	4		10	0	no	(n/a)	(n/a)	Null	materiaPrima

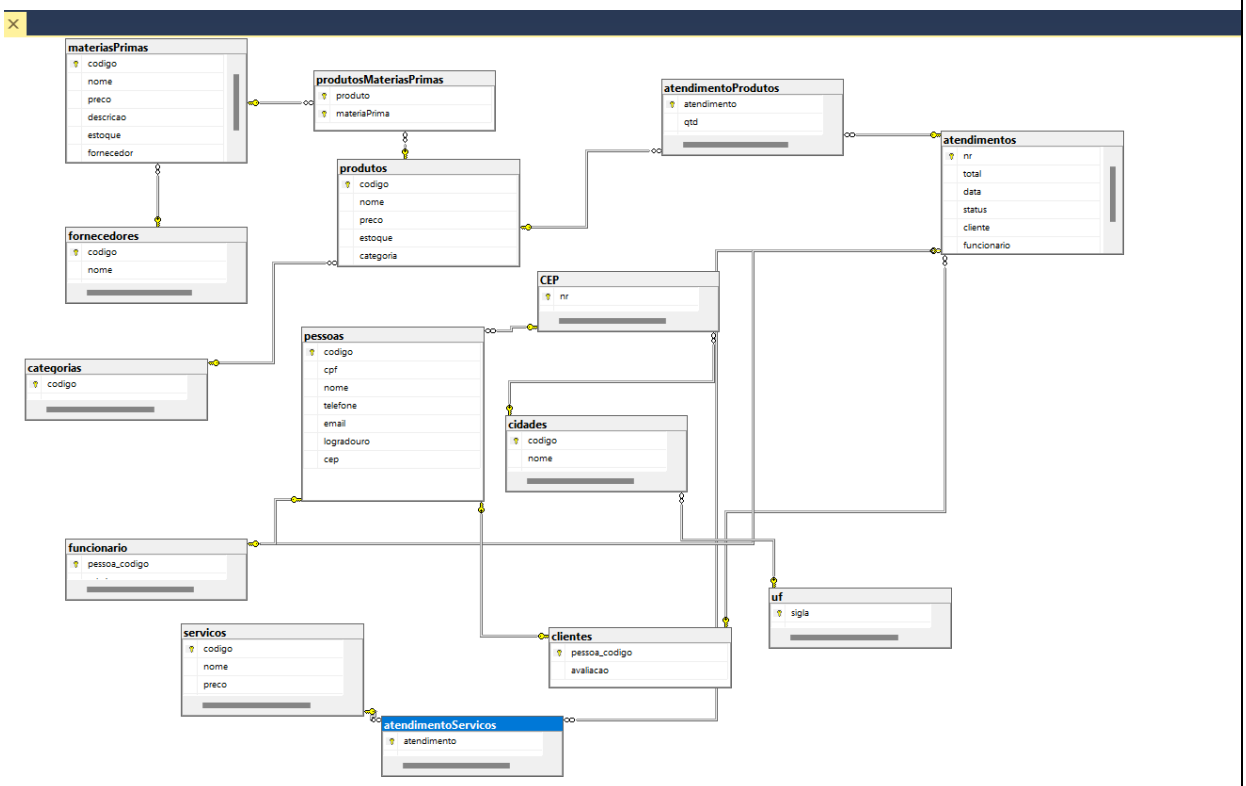
Tabela	Atendimentos									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
nr	int	4		10	0	no	(n/a)	(n/a)	Null	nr
total	decimal	5		7	2	no	(n/a)	(n/a)	Null	total
data	datetime	8				no	(n/a)	(n/a)	Null	data
status	int	4		10	0	yes	(n/a)	(n/a)	Null	status
cliente	int	4		10	0	no	(n/a)	(n/a)	Null	cliente
funcionario	int	4		10	0	no	(n/a)	(n/a)	Null	funcionario

Tabela	Serviços									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
codigo	int	4		10	0	no	(n/a)	(n/a)	Null	codigo
nome	varchar	50				no	no	no	Null	nome
preco	decimal	5		7	2	no	(n/a)	(n/a)	Null	preco
descricao	varchar	50				no	no	no	Null	descricao

Tabela	AtendimentoProduto									
Colunas	**Tipo**	**Tamanho**	**Precisão**	**Escala**	**Null**	**PK**	**FK**	**UK**	**Referência**	**Check**
atendimento	int	4		10	0	no	(n/a)	(n/a)	Null	atendimento
qtd	int	4		10	0	no	(n/a)	(n/a)	Null	qtd
produto	int	4		10	0	no	(n/a)	(n/a)	Null	produto

Tabela	Categorias									
atendimento	int	4		10	0	no	(n/a)	(n/a)	Null	Atendimento
servico	int	4		10	0	no	(n/a)	(n/a)	Null	Serviço
atendimento	int	4		10	0	no	(n/a)	(n/a)	Null	Atendimento

4.4 Diagrama de Banco de Dados



4.5 Script de Banco de Dados

```
       create database sistema_marcenaria3;
go

use sistema_marcenaria3;
go


create table uf (
    sigla varchar(5) not null,
    nome varchar(30) not null,

    constraint pk_uf primary key (sigla)
);
go

create table cidades (
    codigo int not null,
    nome varchar(30) not null,
    sigla_uf varchar(5) not null,

    constraint pk_cidade primary key(codigo),
    constraint fk_ufCidade foreign key(sigla_uf) references uf(sigla)
);
go

create table CEP (
    nr varchar(10) not null,
    codigo_cidade int not null,

    constraint pk_cep primary key (nr),
    constraint fk_cepCidades foreign key (codigo_cidade) references cidades(codigo)
);
go

CREATE TABLE pessoas (
    codigo INT IDENTITY(1,1) NOT NULL,
    cpf VARCHAR(11) NOT NULL,
    nome VARCHAR(30) NOT NULL,
    telefone VARCHAR(15) NULL,
    email VARCHAR(100) NULL,
    logradouro VARCHAR(100) NOT NULL,
    cep VARCHAR(10) NOT NULL,

    CONSTRAINT pk_pessoas PRIMARY KEY (codigo),
    CONSTRAINT fk_pessoasCep FOREIGN KEY (cep) REFERENCES CEP(nr)
);
go

create table clientes (
    pessoa_codigo int not null,
    avaliacao varchar(50) not null,

    constraint pk_clientes primary key (pessoa_codigo),
    foreign key (pessoa_codigo) references pessoas(codigo)
);
go

create table funcionario (
    pessoa_codigo int not null,
    salario decimal(10,2) not null,
```

```
    constraint pk_funcionario primary key (pessoa_codigo),
    foreign key (pessoa_codigo) references pessoas(codigo)
);
go


create table categorias (
    codigo int not null,
    nome varchar(50) not null,

    constraint pk_categorias primary key (codigo)
);
go

create table fornecedores (
    codigo int not null,
    nome varchar(50) not null,
    telefone varchar(15) not null,
    constraint pk_fornecedores primary key (codigo)
);
go

create table materiasPrimas (
    codigo int not null,
    nome varchar(50) not null,
    preco decimal(10,2) not null,
    descricao varchar(100) not null,
    estoque int not null,
    fornecedor int not null,

    constraint pk_materiasPrimas primary key (codigo),
    constraint fk_materiasFornecedor foreign key (fornecedor) references
fornecedores(codigo)
);
go

create table produtos (
    codigo int IDENTITY(1,1) not null,
    nome varchar(50) not null,
    preco decimal(10,2) not null,
    estoque int not null,
    categoria int not null,

    constraint pk_produtos primary key(codigo),
    constraint fk_produtoCategoria foreign key(categoria) references categorias(codigo)
);
go

create table produtosMateriasPrimas (
    produto int not null,
    materiaPrima int not null,

    constraint pk_produtosMateriasPrimas primary key (produto, materiaPrima),
    foreign key (produto) references produtos(codigo),
    foreign key (materiaPrima) references materiasPrimas(codigo)
);
go

CREATE TABLE atendimentos (
    nr int IDENTITY(1,1) NOT NULL,
    total decimal(10,2) NOT NULL,
    data DATETIME NOT NULL,
    status INT CHECK(status IN (1,2,3,4)),
    cliente INT NOT NULL,
```

```
    funcionario INT NOT NULL,

    CONSTRAINT pk_atendimento PRIMARY KEY (nr),
    CONSTRAINT fk_atendimentoCliente FOREIGN KEY (cliente) REFERENCES
clientes(pessoa_codigo),
    CONSTRAINT fk_atendimentoFuncionario FOREIGN KEY (funcionario) REFERENCES
funcionario(pessoa_codigo)
);
go

create table servicos (
    codigo int IDENTITY(1,1) not null,
    nome varchar(50) not null,
    preco decimal(10,2) not null,
    descricao varchar(100) not null,

    constraint pk_servico primary key (codigo)
);
go

create table atendimentoProdutos (
    atendimento int not null,
    qtd int not null,
    produto int not null,

    constraint pk_atendimentoProduto primary key (atendimento, produto),
    foreign key (atendimento) references atendimentos(nr),
    foreign key (produto) references produtos(codigo)
);
go

create table atendimentoServicos (
    atendimento int not null,
    servico int not null,

    constraint pk_atendimentoServicos primary key (atendimento, servico),
    foreign key (atendimento) references atendimentos(nr),
    foreign key (servico) references servicos(codigo)
);
go
```

4.6 Procedures

```
CREATE PROCEDURE Inserir_UF
    @sigla VARCHAR(5),
    @nome VARCHAR(30)
AS
BEGIN
    INSERT INTO uf (sigla, nome)
    VALUES (@sigla, @nome);
END;
GO

---Exemplo---
EXEC Inserir_UF 'SP', 'São Paulo';
EXEC Inserir_UF 'RJ', 'Rio de Janeiro';
EXEC Inserir_UF 'MG', 'Minas Gerais';
```

```
CREATE PROCEDURE Inserir_Cidade
    @codigo INT,
    @nome VARCHAR(30),
    @sigla_uf VARCHAR(5)
AS
BEGIN
    INSERT INTO cidades (codigo, nome, sigla_uf)
    VALUES (@codigo, @nome, @sigla_uf);
END;
GO

---Exemplo---
EXEC Inserir_Cidade 1, 'São Paulo', 'SP';
EXEC Inserir_Cidade 2, 'Rio de Janeiro', 'RJ';
EXEC Inserir_Cidade 3, 'Belo Horizonte', 'MG';



CREATE PROCEDURE Inserir_CEP
    @nr VARCHAR(10),
    @codigo_cidade INT
AS
BEGIN
    INSERT INTO CEP (nr, codigo_cidade)
    VALUES (@nr, @codigo_cidade);
END;
GO

---Exemplo---
EXEC Inserir_CEP '01000-000', 1;
EXEC Inserir_CEP '20000-000', 2;
EXEC Inserir_CEP '30000-000', 3;



CREATE PROCEDURE Inserir_Cliente
    @pessoa_codigo INT,
    @avaliacao VARCHAR(50)
AS
BEGIN
    INSERT INTO clientes (pessoa_codigo, avaliacao)
    VALUES (@pessoa_codigo, @avaliacao);
END;
GO

---Exemplo---
EXEC Inserir_Cliente 1, 'Cliente 1 - Boa avaliação';
EXEC Inserir_Cliente 2, 'Cliente 2 - Avaliação média';
EXEC Inserir_Cliente 3, 'Cliente 3 - Excelente cliente';



CREATE PROCEDURE Inserir_Funcionario
    @pessoa_codigo INT,
    @salario DECIMAL(10,2)
AS
BEGIN
    INSERT INTO funcionario (pessoa_codigo, salario)
    VALUES (@pessoa_codigo, @salario);
END;
GO

---Exemplo---
EXEC Inserir_Funcionario 1, 3000.00;
EXEC Inserir_Funcionario 2, 2500.00;
```

```
EXEC Inserir_Funcionario 3, 3500.00;



CREATE PROCEDURE Inserir_Categoria
    @codigo INT,
    @nome VARCHAR(50)
AS
BEGIN
    INSERT INTO categorias (codigo, nome)
    VALUES (@codigo, @nome);
END;
GO

---Exemplo---
EXEC Inserir_Categoria 1, 'Móveis de Madeira';
EXEC Inserir_Categoria 2, 'Móveis de Metal';
EXEC Inserir_Categoria 3, 'Móveis de Vidro';



CREATE PROCEDURE Inserir_Fornecedor
    @codigo INT,
    @nome VARCHAR(50),
    @telefone VARCHAR(15)
AS
BEGIN
    INSERT INTO fornecedores (codigo, nome, telefone)
    VALUES (@codigo, @nome, @telefone);
END;
GO

---Exemplo---
EXEC Inserir_Fornecedor 1, 'Fornecedor A', '123456789';
EXEC Inserir_Fornecedor 2, 'Fornecedor B', '987654321';
EXEC Inserir_Fornecedor 3, 'Fornecedor C', '999999999';



CREATE PROCEDURE Inserir_MateriaPrima
    @codigo INT,
    @nome VARCHAR(50),
    @preco DECIMAL(10,2),
    @descricao VARCHAR(100),
    @estoque INT,
    @fornecedor INT
AS
BEGIN
    INSERT INTO materiasPrimas (codigo, nome, preco, descricao, estoque, fornecedor)
    VALUES (@codigo, @nome, @preco, @descricao, @estoque, @fornecedor);
END;
GO

---Exemplo---
EXEC Inserir_MateriaPrima 1, 'Madeira', 50.00, 'Madeira de alta qualidade', 100, 1;
EXEC Inserir_MateriaPrima 2, 'Metal', 30.00, 'Metal resistente', 150, 2;
EXEC Inserir_MateriaPrima 3, 'Vidro', 40.00, 'Vidro temperado', 80, 3;

CREATE PROCEDURE Inserir_Produto
    @nome VARCHAR(50),
    @preco DECIMAL(10,2),
    @estoque INT,
    @categoria INT
AS
BEGIN
```

```
    INSERT INTO produtos (nome, preco, estoque, categoria)
    VALUES (@nome, @preco, @estoque, @categoria);
END;
GO

---Exemplo---
EXEC Inserir_Produto 'Mesa de Madeira', 300.00, 10, 1;
EXEC Inserir_Produto 'Cadeira de Metal', 150.00, 20, 2;
EXEC Inserir_Produto 'Armário de Vidro', 500.00, 5, 3;



CREATE PROCEDURE Inserir_ProdutoMateriaPrima
    @produto INT,
    @materiaPrima INT
AS
BEGIN
    INSERT INTO produtosMateriasPrimas (produto, materiaPrima)
    VALUES (@produto, @materiaPrima);
END;
GO

---Exemplo---
EXEC Inserir_ProdutoMateriaPrima 1, 1;
EXEC Inserir_ProdutoMateriaPrima 2, 2;
EXEC Inserir_ProdutoMateriaPrima 3, 3;



CREATE PROCEDURE Inserir_Atendimento
    @total DECIMAL(10,2),
    @data DATETIME,
    @status INT,
    @cliente INT,
    @funcionario INT
AS
BEGIN
    INSERT INTO atendimentos (total, data, status, cliente, funcionario)
    VALUES (@total, @data, @status, @cliente, @funcionario);
END;
GO

---Exemplo---
EXEC Inserir_Atendimento 500.00, '2024-05-21 10:00:00', 1, 1, 1;
EXEC Inserir_Atendimento 800.00, '2024-05-22 11:00:00', 2, 2, 2;
EXEC Inserir_Atendimento 1000.00, '2024-05-23 12:00:00', 3, 3, 3;



CREATE PROCEDURE Inserir_Servico
    @nome VARCHAR(50),
    @preco DECIMAL(10,2),
    @descricao VARCHAR(100)
AS
BEGIN
    INSERT INTO servicos (nome, preco, descricao)
    VALUES (@nome, @preco, @descricao);
END;
GO

---Exemplo---
EXEC Inserir_Servico 'Montagem de Móveis', 100.00, 'Montagem profissional de móveis';
EXEC Inserir_Servico 'Reparo de Móveis', 80.00, 'Reparos em móveis danificados';
```

```
EXEC Inserir_Servico 'Consultoria de Decoração', 150.00, 'Consultoria especializada em
decoração de interiores';


CREATE PROCEDURE Inserir_AtendimentoServico
    @atendimento INT,
    @servico INT
AS
BEGIN
    INSERT INTO atendimentoServicos (atendimento, servico)
    VALUES (@atendimento, @servico);
END;
GO

---Exemplo---
EXEC Inserir_AtendimentoServico 1, 1;
EXEC Inserir_AtendimentoServico 2, 2;
EXEC Inserir_AtendimentoServico 3, 3;



CREATE PROCEDURE Inserir_AtendimentoProduto
    @atendimento INT,
    @qtd INT,
    @produto INT
AS
BEGIN
    INSERT INTO atendimentoProdutos (atendimento, qtd, produto)
    VALUES (@atendimento, @qtd, @produto);
END;
GO

---Exemplo---
EXEC Inserir_AtendimentoProduto 1, 2, 1;
EXEC Inserir_AtendimentoProduto 2, 3, 2;
EXEC Inserir_AtendimentoProduto 3, 1, 3;
```

4.7 Views

1. View de Informações do Cliente:

```
CREATE VIEW vw_info_clientes AS
SELECT
    p.codigo AS codigo_cliente,
    p.nome AS nome_cliente,
    p.telefone AS telefone_cliente,
    p.email AS email_cliente,
    p.logradouro AS logradouro_cliente,
    c.nome AS cidade_cliente,
    u.nome AS uf_cliente,
    ce.nr AS cep_cliente,
    cli.avaliacao AS avaliacao_cliente
FROM
    pessoas p
    JOIN cidades c ON p.cep = c.codigo
    JOIN ufs u ON c.sigla_uf = u.sigla
    JOIN CEPs ce ON c.codigo = ce.codigo_cidade
    JOIN clientes cli ON p.codigo = cli.pessoa_codigo;
```

2. View de Informações de Funcionários:

```
CREATE VIEW vw_info_funcionarios AS
SELECT
    p.codigo AS codigo_funcionario,
    p.nome AS nome_funcionario,
    p.telefone AS telefone_funcionario,
    p.email AS email_funcionario,
    p.logradouro AS logradouro_funcionario,
    c.nome AS cidade_funcionario,
    u.nome AS uf_funcionario,
    ce.nr AS cep_funcionario,
    f.salario AS salario_funcionario
FROM
    pessoas p
    JOIN cidades c ON p.cep = c.codigo
    JOIN ufs u ON c.sigla_uf = u.sigla
    JOIN CEPs ce ON c.codigo = ce.codigo_cidade
    JOIN funcionario f ON p.codigo = f.pessoa_codigo;
```

3. View de Informações de Produtos:

```
CREATE VIEW vw_info_produtos AS
SELECT
    pr.codigo AS codigo_produto,
    pr.nome AS nome_produto,
    pr.preco AS preco_produto,
    pr.estoque AS estoque_produto,
    c.nome AS categoria_produto
FROM
    produtos pr
    JOIN categorias c ON pr.categoria = c.codigo;
```

4. View de Informações de Atendimentos:

```
CREATE VIEW vw_info_atendimentos AS
SELECT
    a.nr AS numero_atendimento,
    a.total AS total_atendimento,
    a.data AS data_atendimento,
    CASE
        WHEN a.status = 1 THEN 'Aberto'
        WHEN a.status = 2 THEN 'Em andamento'
        WHEN a.status = 3 THEN 'Concluído'
        WHEN a.status = 4 THEN 'Cancelado'
    END AS status_atendimento,
    CONCAT(c.nome, ' - ', p.nome) AS cliente_atendimento,
    CONCAT(f.pessoa_codigo, ' - ', p.nome) AS funcionario_atendimento
FROM
    atendimentos a
    JOIN clientes cli ON a.cliente = cli.pessoa_codigo
    JOIN pessoas p ON cli.pessoa_codigo = p.codigo
    JOIN cidades c ON p.cep = c.codigo
    JOIN funcionario f ON a.funcionario = f.pessoa_codigo;
```

5. Conclusão

A realização deste projeto foi um exercício valioso para nossa equipe, composta por seis membros: Matheus, Vinicius, Iago, Felipe, Eduardo e Túlio. Cada um teve um papel crucial no desenvolvimento do sistema MarcenariaPro, enfrentando desafios específicos que contribuíram para nosso aprendizado coletivo.

Matheus, responsável pelo Diagrama de Classe/Modelo Conceitual, inicialmente enfrentou dificuldades na implementação da parte de atendimento/serviços. Apesar disso, conseguiu superar essa barreira e, após relembrar como construir diagramas de classe, finalizou a tarefa com sucesso.

Vinicius, encarregado do Modelo Lógico, encontrou dificuldades na utilização das chaves primárias e estrangeiras. Após pesquisas e análise de trabalhos anteriores, ele conseguiu aplicar corretamente esses conceitos no nosso projeto.

Tulio, responsável pelo script, um pouco de dificuldade em traduzir o mapeamento porém após conversa com pessoal do grupo conseguiu resolver.

Iago, que cuidou da Introdução e Requisitos, teve dificuldades em descrever os comportamentos no modelo da documentação, mas conseguiu completar a tarefa sem maiores problemas.

Felipe, responsável pelas Views, inicialmente não tinha conhecimento sobre o assunto. No entanto, ele se aprofundou na temática e conseguiu entender e aplicar as views no nosso sistema.

Eduardo, que trabalhou nas Procedures, precisou recorrer a videoaulas e realizar vários testes para compreender o funcionamento dessas rotinas. Seu esforço resultou na implementação correta das procedures necessárias.

Esses desafios enfrentados e superados por cada membro demonstram a importância do trabalho em equipe e da dedicação individual. O projeto MarcenariaPro não só atendeu aos requisitos propostos, mas também proporcionou um grande aprendizado para todos os envolvidos. Agradecemos ao professor Sérgio Borges pela orientação e estamos confiantes de que este trabalho contribuirá positivamente para o mercado de madeireiras, oferecendo uma solução eficiente e personalizada.