



Dr. Tok

O Tok da Limpeza

André Couto
Igor Pereira
Guilherme Panetto
Rafael Guimarães

Sumário

- **Introdução**
 - Definição do Tema
- **Especificação de Requisitos**
 - Descrição Geral
- **Requisitos**
 - Requisitos de Hardware
 - Requisitos de Software
- **Modelagem**
 - MER
 - Modelo Lógico e Físico
 - Dicionário de Dados
 - Restrição de Integridade
- **Scripts**

Introdução

Definição do Tema

- Empresa de Terceirização de Serviços de Limpeza
 - Contratação dos Serviços
 - Funcionários Especializados
 - Produtos e Materiais de Limpeza



Especificação de Requisitos

Descrição Geral

- Descrição Geral
 - Pontos identificados e escolhidos para a base do sistema:
 - Funcionário Administrativo
 - Funcionário Externo
 - Salário
 - Contracheque
 - Alocação
 - Estoque
 - Serviço
 - Fornecedor
 - Entrega
 - Cliente
 - Contrato
 - Produto



Descrição Geral

- **Funções do Produto**
 - Gerenciamento de contratos
 - Gravando detalhes do Cliente
 - Atualização do registro com contrato
 - Impressão de despesas
 - Renovação de contrato
 - Sistema de gerenciamento de funcionários e salários
 - Atendimento e horário dos funcionários
 - Aprovações de férias
 - Horário de trabalho
 - Cálculo impostos

Descrição Geral

- **Funções do Produto**
 - Gerenciamento de Estoque
 - Gerenciamento de estoque
 - Notificação de vencimento
 - Notificação de falta de produto
 - Cálculo de faturamento

Descrição Geral

- **Características do Usuário**
 - Administrador
 - Gerente Comercial
 - Gerente Financeiro
 - Gerente Recursos Humanos
 - Gerente Operacional
 - Sistema

Requisitos

Requisitos de Hardware

- **Interfaces de Hardware**
 - Notebook/Desktop PC
 - Processador - Core i3
 - RAM - 4GB
 - HDD - 500GB
 - LED/LCD Monitor/TV
 - Impressora Laser (P/B)
 - Roteador Wi-Fi

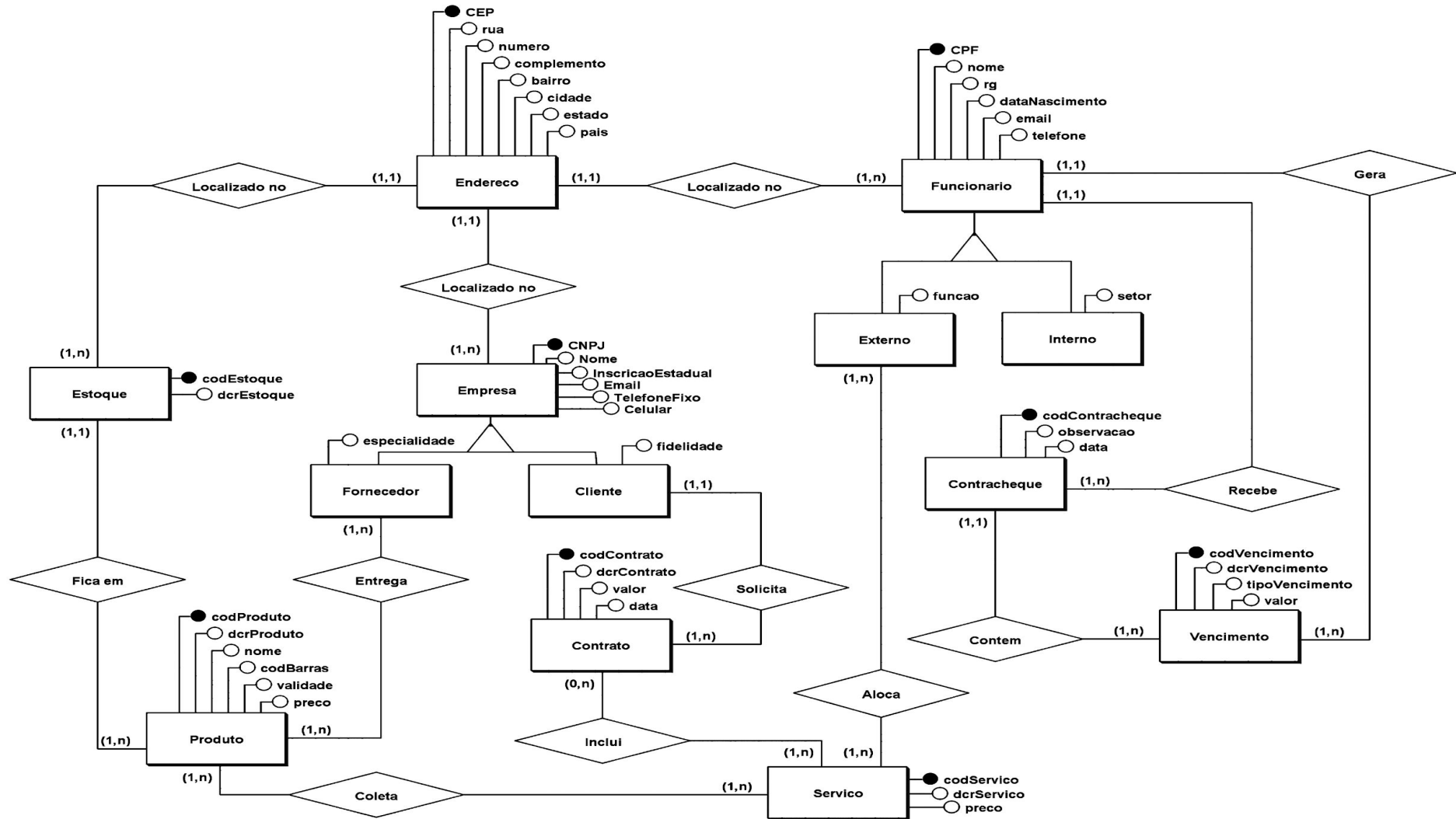
Requisitos de Software

- **Interfaces de Software**

- OS
 - Windows 7 ou posterior, Linux Mint ou equivalente.
- JRE 1.8
 - JAVA Runtime Environment para uma possível implementação gráfica do sistema
- MySQL server
 - Conexão com o banco de Dados.

Modelagem

Modelo de Entidades e Relacionamentos



Modelo Lógico

Modelo Físico

```
1 DROP SCHEMA IF EXISTS `DRTOK`;
2 CREATE SCHEMA `DRTOK`;
3 USE `DRTOK`;
4
5 CREATE TABLE Endereco (
6     CEP INT NOT NULL,
7     rua VARCHAR(50) NOT NULL,
8     numero INT NOT NULL,
9     complemento VARCHAR(150) NOT NULL,
10    bairro VARCHAR(50) NOT NULL,
11    cidade VARCHAR(50) NOT NULL,
12    estado VARCHAR(50) NOT NULL,
13    pais VARCHAR(50) NOT NULL,
14    PRIMARY KEY (CEP)
15 );
16
17 CREATE TABLE Funcionario (
18     CPF CHAR(11) NOT NULL,
19     nome VARCHAR(50) NOT NULL,
20     rg CHAR(9) NOT NULL,
21     dataNascimento DATE NOT NULL,
22     email VARCHAR(100) NOT NULL,
23     telefone CHAR(12) NOT NULL,
24     funcao VARCHAR(150) NOT NULL,
25     setor VARCHAR(50) NOT NULL,
26     tipoFuncionario INT NOT NULL CHECK(tipoFuncionario in (1,2)),
27     FKEnderecoCEP INT NOT NULL,
28     PRIMARY KEY (CPF),
29     CONSTRAINT FKFuncionario FOREIGN KEY (FKEnderecoCEP)
30         REFERENCES Endereco (CEP) ON DELETE RESTRICT ON UPDATE RESTRICT
31 );
```

Dicionário de Dados

Tabela	Empresa			
Descrição	Organização econômica, civil ou comercial, constituída para explorar um ramo de negócio, podendo ser cliente ou fornecedor			
Observações	Possui chave estrangeira da tabela Endereço			
Campos				
Nome	Descrição	Tipo	Tamanho	Restrições
CNPJ	Número único que identifica uma pessoa jurídica	char	14	NOT NULL
Nome	Nome da empresa	varchar	50	NOT NULL
InscricaoEstadual	Registro formal do negócio junto à Receita Estadual.	char	9	NOT NULL
Email	Endereço eletrônico da empresa	varchar	150	NOT NULL
TelefoneFixo	Telefone de contato fixo da empresa	char	12	NOT NULL
Celular	Telefone de contato móvel da empresa	char	12	NOT NULL

Tabela	Fornecedor			
Descrição	Organização que fornece produtos para realização do serviço			
Observações	Tabela derivada de Empresa			
Campos				
Nome	Descrição	Tipo	Tamanho	Restrições
especialidade	Especialidade do fornecedor	varchar	100	NOT NULL

Tabela	Cliente			
Descrição	Organização que por meio de um contrato adquire serviço da empresa			
Observações	Tabela derivada de Empresa			
Campos				
Nome	Descrição	Tipo	Tamanho	Restrições
fidelidade	Quando o primeiro contrato foi estabelecido	date		NOT NULL

Tabela	Funcionario				
Descrição	Mão de obra assalariada contratada pela empresa para prestação de serviços				
Observações	Possui chave estrangeira da tabela Endereço				
Campos					
Nome	Descrição	Tipo	Tamanho	Restrições	
CPF	Número único que identifica um a pessoa física	char	11	NOT NULL	
nome	Nome do funcionário	varchar	50	NOT NULL	
rg	Documento nacional de identificação civil	char	9	NOT NULL	
dataNascimento	Data de nascimento do funcionário	date		NOT NULL	
email	Endereço eletrônico do funcionário	varchar	100	NOT NULL	
telefone	Telefone para contato do funcionário	char	12	NOT NULL	

Tabela	Produto			
Descrição	Um bem adquirido de um fornecedor, que é utilizado para a execução dos serviços pelos funcionários.			
Observações	Possui uma chave estrangeira da tabela Estoque			
Campos				
Nome	Descrição	Tipo	Tamanho	Restrições
codProduto	Código do produto	int		NOT NULL
dcrProduto	Descrição do produto	varchar	150	NOT NULL
nome	Nome do produto	varchar	150	NOT NULL
codBarras	Representação gráfica de dados numéricos do produto	int		NOT NULL
validade	Data de validade do produto	date		NOT NULL
preco	Valor do produto	numeric	10.2	NOT NULL

Restrições de Integridade

Restrições de Integridades

- Restrições Identificadas

- Na tabela de Produto, uma data de validade não pode ser inferior a data presente no momento da criação do registro;
- Na tabela Vencimento, o tipo de vencimento pode assumir apenas 2 valores, sendo eles receita e desconto;
- Na tabela Empresa o tipo de empresa pode assumir apenas 2 valores, sendo eles 1 e 2;
- Na tabela de Contracheque, uma data não pode ser inferior a data presente no momento da criação do registro;
- Na tabela de Vencimento, um valor não pode ser menor que 0 (zero);

Scripts

Script de Criação

```
1 DROP SCHEMA IF EXISTS `DRTOK`;
2 CREATE SCHEMA `DRTOK`;
3 USE `DRTOK`;
4
5 CREATE TABLE Endereco (
6     CEP INT NOT NULL,
7     rua VARCHAR(50) NOT NULL,
8     numero INT NOT NULL,
9     complemento VARCHAR(150) NOT NULL,
10    bairro VARCHAR(50) NOT NULL,
11    cidade VARCHAR(50) NOT NULL,
12    estado VARCHAR(50) NOT NULL,
13    pais VARCHAR(50) NOT NULL,
14    PRIMARY KEY (CEP)
15 );
16
17 CREATE TABLE Funcionario (
18     CPF CHAR(11) NOT NULL,
19     nome VARCHAR(50) NOT NULL,
20     rg CHAR(9) NOT NULL,
21     dataNascimento DATE NOT NULL,
22     email VARCHAR(100) NOT NULL,
23     telefone CHAR(12) NOT NULL,
24     funcao VARCHAR(150) NOT NULL,
25     setor VARCHAR(50) NOT NULL,
26     tipoFuncionario INT NOT NULL CHECK(tipoFuncionario in (1,2)),
27     FKEnderecoCEP INT NOT NULL,
28     PRIMARY KEY (CPF),
29     CONSTRAINT FKFuncionario FOREIGN KEY (FKEnderecoCEP)
30         REFERENCES Endereco (CEP) ON DELETE RESTRICT ON UPDATE RESTRICT
31 );
```

Script de Carga de Dados

```
1 ## Insert
2 ##### Inserts, a Carga de dados para fazer o Banco de dados Girar, e possibilitar a construção de relações
  de dados reais
3
4 USE `DRTOK`;
5
6 INSERT INTO
7 Vencimento (dcrVencimento, valor, tipoVencimento, FKContrachequeCodContracheque, FKFuncionarioCPF) VALUES
8   ("SALÁRIO BASE", 5892.43, "RECEITA", 1, "77209625755"),
9   ("SALÁRIO FAMÍLIA", 647.83, "RECEITA", 1, "77209625755"),
10  ("FÉRIAS", 591.42, "RECEITA", 1, "77209625755"),
11  ("GRATIFICAÇÃO", 538.90, "RECEITA", 1, "77209625755"),
12  ("BONIFICAÇÃO", 883.61, "RECEITA", 1, "77209625755"),
13  ("HORAS EXTRAS", 719.41, "RECEITA", 1, "77209625755"),
14  ("ADICIONAL INSALUBRIDADE", 818.27, "RECEITA", 1, "77209625755"),
15  ("ADICIONAL NOTURNO", 920.54, "RECEITA", 1, "77209625755"),
16  ("ADICIONAL PONTUALIDADE", 188.46, "RECEITA", 1, "77209625755"),
17  ("AUXILIO ALIMENTAÇÃO", 472.95, "RECEITA", 1, "77209625755"),
18  ("FALTAS", 507.97, "DESCONTO", 1, "77209625755"),
19  ("INSS", 406.74, "DESCONTO", 1, "77209625755"),
20  ("VALE TRANSPORTE", 989.54, "DESCONTO", 1, "77209625755"),
21  ("CONVÊNIO", 864.07, "DESCONTO", 1, "77209625755"),
22  ("ADIANTAMENTO QUINZENAL", 664.68, "DESCONTO", 1, "77209625755"),
23  ("CONTRIBUIÇÃO SINDICAL", 384.74, "DESCONTO", 1, "77209625755"),
24  ("SALÁRIO BASE", 3461.48, "RECEITA", 2, "31906491704"),
25  ("SALÁRIO FAMÍLIA", 580.76, "RECEITA", 2, "31906491704"),
26  ("FÉRIAS", 973.56, "RECEITA", 2, "31906491704"),
27  ("GRATIFICAÇÃO", 861.50, "RECEITA", 2, "31906491704"),
```

Scripts de View

```
1 ## Visões
2 ### Identificar partes do modelo bastante suscetíveis a consultas
3 ### Implementar pelo menos duas views para as partes do modelo
4
5 USE `DRTOK`;
6
7 #####
8 # 1 A descoberta dos produtos (ou localizados em um estoque),
9 # utilizado para fazer um pedido de reposição de produtos no estoque
10
11 DROP VIEW IF EXISTS Suprimentos;
12 CREATE VIEW Suprimentos AS
13     SELECT p.codproduto as Código,
14            p.nome as Nome,
15            e.dcrestoque as Estoque,
16            p.dcrproduto as Descrição,
17            p.preco as Preço,
18            p.validade as Validade
19     FROM Produto as p, Estoque as e
20     -- 1/2 Opcionalmente podemos modificar a linha para conseguir
21     -- produtos de todos os estoques
22     -- WHERE p.fkestoquecodestoque = (
23     --     SELECT es.codestoque
24     --     FROM estoque as es, endereco as e
25     --     WHERE es.fkenderecocep = e.cep AND e.cidade = "colatina")
26     WHERE p.fkestoquecodestoque = e.codestoque
27     ORDER BY e.dcrestoque;
28
29 SELECT * FROM Suprimentos;
30
```

Scripts de View

```
31 #####
32 # 2 Descoberta de todos os produtos de um serviço e com a identificação do Fornecedor
33
34 DROP VIEW IF EXISTS FornecedorProdutoServico;
35 CREATE VIEW FornecedorProdutoServico AS
36     SELECT  s.dcrServico as Serviço,
37             p.codProduto as Código,
38             p.nome as Produto,
39             em.nome as Fornecedor
40     FROM Produto as p, Entrega as e, coleta as c, servico as s, estoque as es, empresa as em
41     WHERE em.cnpj = e.fkempresacnpj
42           AND p.codProduto = e.FKProdutoCodProduto
43           AND p.codProduto = c.fkprodutocodproduto
44           AND c.fkservicocodservico = s.codServico
45           AND p.fkestoquecodestoque = es.codEstoque
46     ORDER BY s.codServico, em.cnpj;
47
48 SELECT * FROM FornecedorProdutoServico;
```

Script de Stored Procedures

```
1 ## Stored Procedures
2 ### Identificar duas situações para implementação de procedures
3 ### Implementar as procedures e os casos de testes das procedures
4
5 USE `DRTOK`;
6
7 #####
8 # Procedure que calcula os ganhos de um funcionário com os descontos
9
10 DROP PROCEDURE IF EXISTS Pagamento;
11 DELIMITER //
12 CREATE PROCEDURE Pagamento(IN nomeFuncionario varchar(50))
13 BEGIN
14     SELECT f.Nome as Nome, f.funcao as Função, (
15         (SELECT SUM(v.valor)
16          FROM vencimento as v, contracheque as c, funcionario as f
17          WHERE v.fkcontrachequecodcontracheque = c.codcontracheque
18              AND v.fkfuncionariocpf = f.cpf
19              AND f.cpf = (
20                  SELECT cpf
21                  FROM Funcionario
22                  WHERE nome = nomeFuncionario) AND v.tipovencimento = 'RECEITA'
23         ) - (SELECT SUM(v.valor)
24          FROM vencimento as v, contracheque as c, funcionario as f
25          WHERE v.fkcontrachequecodcontracheque = c.codcontracheque
26              AND v.fkfuncionariocpf = f.cpf
27              AND f.cpf = (
28                  SELECT cpf
29                  FROM Funcionario
30                  WHERE nome = nomeFuncionario) AND v.tipovencimento = 'DESCONTO'
31         ))
32     as Pagamento
33 FROM vencimento as v, contracheque as c, funcionario as f
34 WHERE v.fkcontrachequecodcontracheque = c.codcontracheque
35     AND v.fkfuncionariocpf = f.cpf
36     AND f.cpf = (
37         SELECT cpf
38         FROM Funcionario
39         WHERE nome = nomeFuncionario)
40 GROUP BY f.nome;
41 END //
42 DELIMITER ;
43
44 CALL Pagamento('Cláudio Thales da Mota');
```


Script de Triggers

```
1 ## Triggers
2 ### Identificar duas situações para implementação de triggers
3 ### Implementar os triggers e os casos de testes dos triggers
4
5 USE `DRTOK`;
6
7 #####
8 # Trigger que trata o caso de restrição de integridade do atributo DATA na tabela Contrato
9
10 DROP TRIGGER IF EXISTS TgrContratoVerificaData;
11 DELIMITER //
12 CREATE TRIGGER TgrContratoVerificaData
13 BEFORE INSERT ON contrato
14 FOR EACH ROW
15 BEGIN
16     IF (CURDATE() > new.data) THEN
17         SET new.data = NULL;
18     END IF;
19 END //
20 DELIMITER ;
```

Script de Triggers

```
23 #####
24 # Acompanha o Trigger TgrContratoVerificaData, no qual este trata
25 ## o erro de inserção, caso a data ou outro campo esteja inválido
26
27 DROP PROCEDURE IF EXISTS spInsereContrato;
28 DELIMITER //
29 CREATE PROCEDURE spInsereContrato ( IN spdcrContrato VARCHAR(50),
30                                     IN spvalor NUMERIC(10,2),
31                                     IN spdata DATE,
32                                     IN spFKEmpresaCNPJ CHAR(14))
33 BEGIN
34     DECLARE EXIT HANDLER FOR SQLSTATE '23000'
35     BEGIN
36         SELECT 'Problemas com a inserção data, Data anterior à hoje!' as ERRO;
37     END;
38
39     INSERT INTO Contrato (dcrContrato, valor, data, FKEmpresaCNPJ) VALUES
40     (spdcrContrato, spvalor, spdata, spFKEmpresaCNPJ);
41
42 END//
43 DELIMITER ;
```

Obrigado pela atenção!
