

TRABALHO PRÁTICO 2

Alex Eduardo Alves dos Santos
Bernardo Souza Scaldaferrri
Bruno Silva Marques
Renato Junio Costa Correia
Pedro Henrique de Oliveira Barbosa

1. Apresentação

O objetivo deste trabalho é projetar e implementar um banco de dados relacional para uma loja de componentes eletrônicos. O projeto desse banco de dados se iniciou com o projeto conceitual que será apresentado a seguir, como um diagrama ER(Entidade Relacionamento), essa parte conceitual descreve de forma precisa o banco de dados para que em seguida, na fase do projeto lógico, esse diagrama fosse mapeado para um esquema relacional que descreve a estrutura de cada uma das relações(tabelas) que compõem o banco de dados e as restrições de integridade.

Após definir o esquema relacional com base no diagrama ER, o banco de dados foi criado e instanciado com dados fictícios que permitem a comparação de desempenhos de diferentes consultas SQL.

2. Diagrama ER (Entidade Relacionamento)

No sistema da loja de componentes eletrônicos será armazenado, para cada cliente, o seu CPF, nome, um ou mais telefones de contato e seu endereço que será composto por estado, cidade, bairro, rua e número da casa. No sistema, cada cliente poderá comprar componentes, sendo que cada componente registrado tem um ID próprio, um nome, marca, além de um atributo para checagem se o componente ainda existe no estoque. Cada componente pertence a uma ou mais categorias e cada categoria tem um ID, nome e período de garantia dos componentes daquela categoria.

Todos os componentes são fornecidos à loja por meio de um fornecedor que é identificado pelo seu CNPJ e possui como atributos cadastrados no sistema o seu endereço (também composto por estado, número, cidade, bairro, rua e número) e um ou mais telefones de contato.

Um componente é enviado para uma transportadora através de um operador interno da loja, e cada transportadora, através de um operador pode ter enviado à ela um ou mais componentes para serem entregues. E, claro, um componente, através de um operador só pode ser enviado à uma transportadora para ser entregue.

O operador da loja é identificado pelo seu CPF e também possui seu turno de trabalho e unidade de operação como atributos. Uma transportadora é

identificada pelo seu CNPJ e possui como atributo o tipo de fretagem, além de seu endereço composto por estado, cidade, bairro, rua e número. Uma mesma transportadora faz entregas para um ou vários clientes e um cliente recebe suas entregas através de uma ou mais transportadoras.

Abaixo segue o esquema Entidade-Relacionamento do sistema proposto.

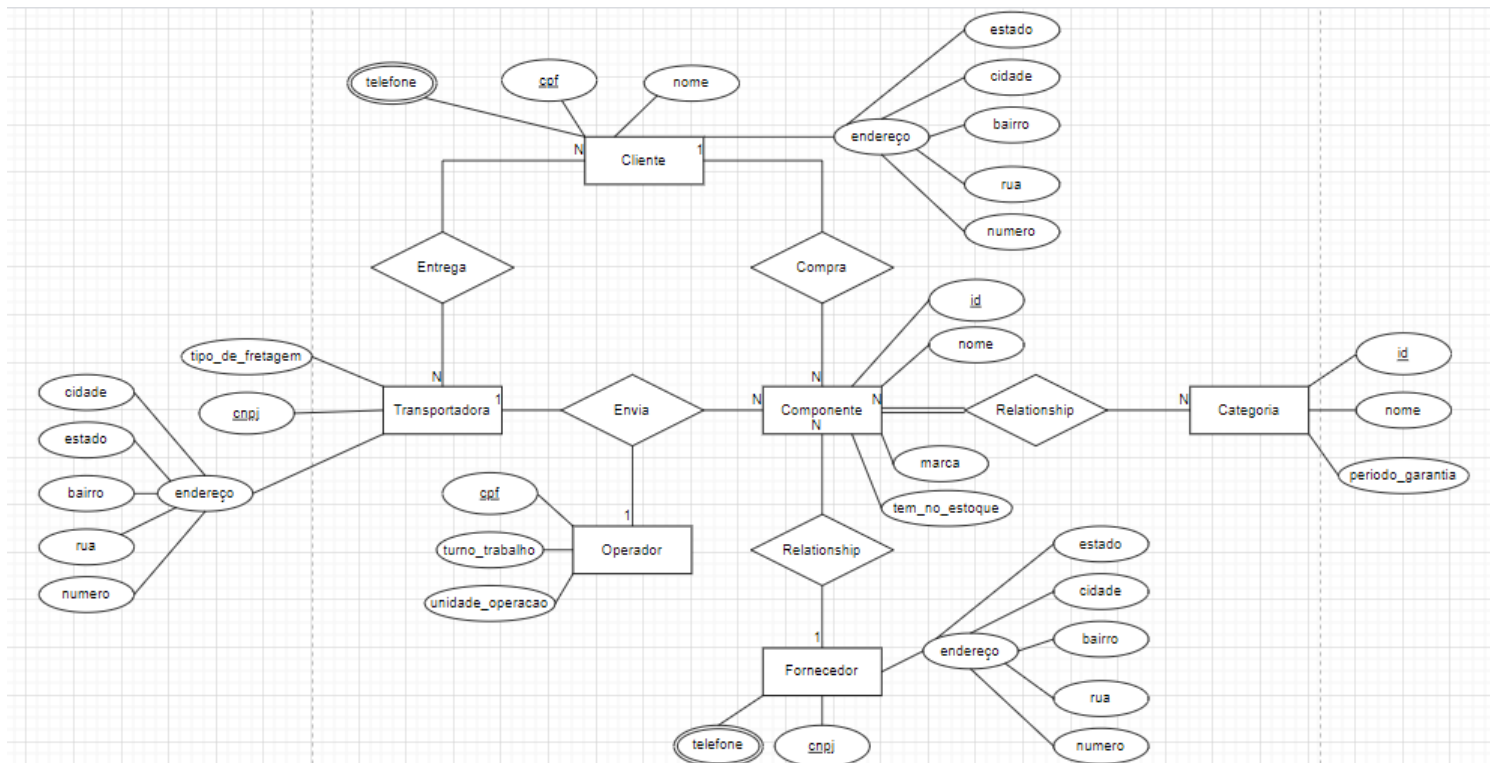


Imagem 1: Diagrama ER do banco de dados de uma loja de componentes eletrônicos.

3. Projeto Lógico

Com base no diagrama apresentado, o projeto lógico a seguir foi criado, o esquema relacional a seguir contém todas as tabelas do banco de dados, assim como suas restrições de integridade, além disso, também foi feito o diagrama UML das tabelas, a fim de ter uma visualização gráfica das relações do banco de dados.

Entidades:

- Cliente

Cliente(CPF(nn), nome)

Telefone-Cliente(CPF(nn), telefone)

Endereco-Cliente(CPF(nn), estado, cidade, bairro, rua, numero)

Telefone-Cliente[CPF] \xrightarrow{p} Cliente[CPF]

Endereco-Cliente[CPF] \xrightarrow{p} Cliente[CPF]

- **Transportadora**

Transportadora(CNPJ(nn), tipo_de_fretagem)

Endereco-Transportadora(CNPJ(nn), estado, cidade, bairro, rua, numero)

Endereco-Transportadora[CNPJ] \xrightarrow{p} Transportadora[CNPJ]

- **Operador**

Operador(CPF(nn), turno_trabalho, unidade_operacao)

- **Fornecedor**

Fornecedor(CNPJ(nn))

Telefone-Fornecedor(CNPJ(nn), telefone)

Endereco-Fornecedor(CNPJ(nn), estado, cidade, bairro, rua, numero)

Telefone-Fornecedor[CNPJ] \xrightarrow{p} Fornecedor[CNPJ]

Endereco-Fornecedor[CNPJ] \xrightarrow{p} Fornecedor[CNPJ]

- **Categoria**

Categoria(ID(nn), nome, periodo_garantia)

- **Componente**

Componente(ID(nn), nome, marca, tem_no_estoque, categoria(nn))

Componente[categoria] \xrightarrow{b} Categoria[ID]

Relacionamentos:

- **Fornece**

Fornece(CNPJ(nn), ID(nn))

Fornece[ID] \xrightarrow{p} Componente[ID]

Fornece[CNPJ] \xrightarrow{p} Fornecedor[CNPJ]

- **Envia**

Envia(CNPJ(nn), CPF(nn), ID(nn))

Envia[ID] \xrightarrow{p} Componente[ID]

Envia[CPF] \xrightarrow{p} Operador[CPF]

Envia[CNPJ] \xrightarrow{p} Transportadora[CNPJ]

- **Entrega**

Entrega(CNPJ(nn), CPF(nn))

Entrega[CPF] \xrightarrow{p} Cliente[CPF]

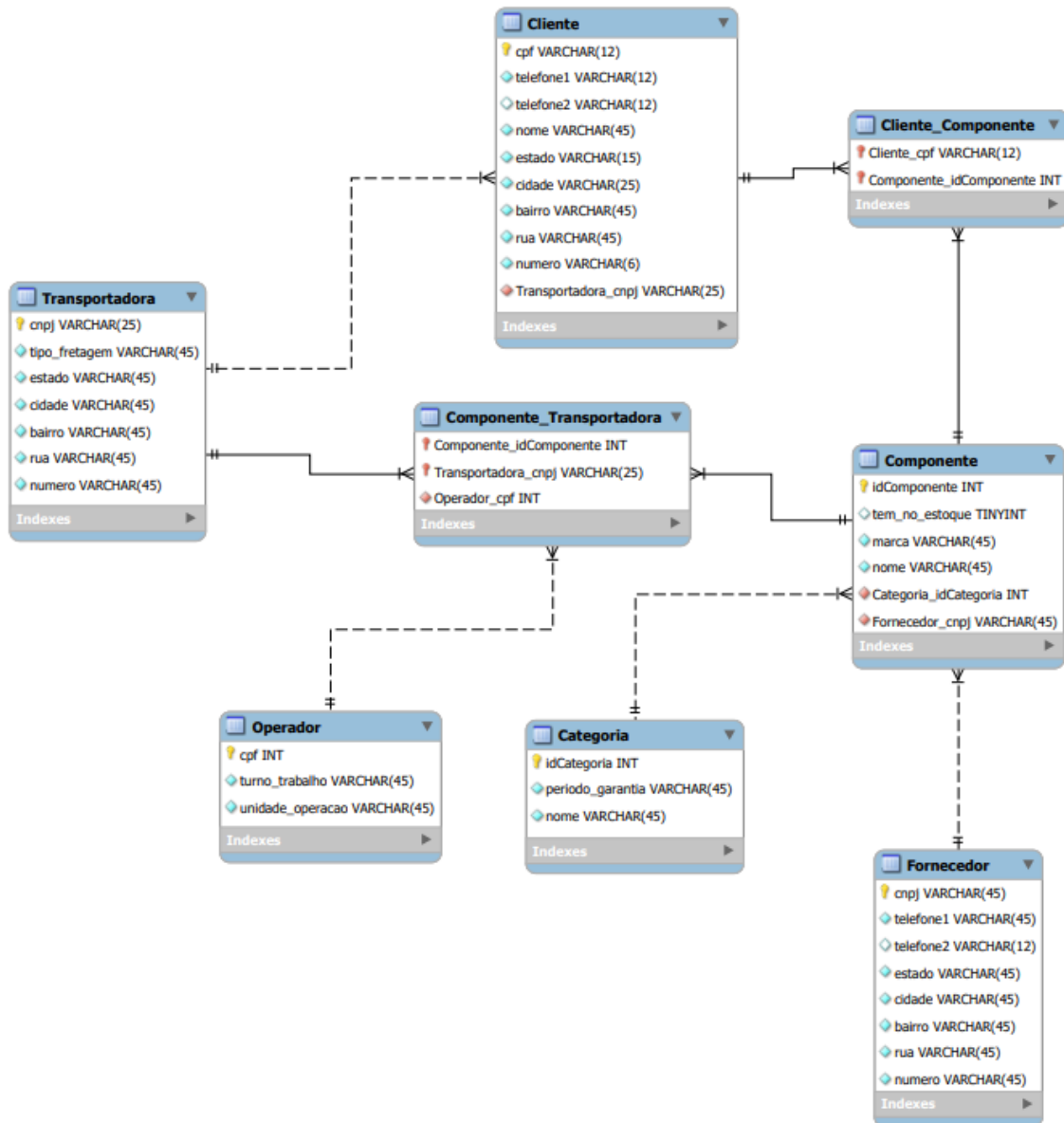
Entrega[CNPJ] \xrightarrow{p} Transportadora[CNPJ]

- **Compra**

Compra(CPF(nn), ID(nn))

Compra[CPF] \xrightarrow{b} Cliente[CPF]

Compra[ID] \xrightarrow{b} Componente[ID]



4. Consultas SQL (Querys)

Após o desenvolvimento do projeto lógico, o banco de dados foi implementado, e foi populado com dados que permitissem testar o desempenho das consultas SQL a seguir:

1. Obter o nome de todos os clientes do estado de SP(exemplo)

```
SELECT nome FROM cliente WHERE estado = 'SP'
```

```
22:04:34      SELECT nome FROM cliente WHERE estado = 'SP'  
LIMIT 0, 1000      0 row(s) returned    0.000 sec / 0.000 sec
```

2. Obter o turno de trabalho de todos os operadores da unidade X

```
SELECT turno_trabalho FROM Operador WHERE unidade_operacao  
= 'My'
```

```
22:06:24      SELECT turno_trabalho FROM Operador WHERE  
unidade_operacao = 'My' LIMIT 0, 1000 5 row(s) returned    0.000 sec  
/ 0.000 sec
```

3. Obter os nome dos componentes oferecidos por fornecedores de Belo Horizonte(exemplo)

```
SELECT nome  
FROM Fornecedor INNER JOIN Componente  
on Fornecedor.cnpj = Componente.Fornecedor_cnpj  
WHERE cidade = "Belo Horizonte"
```

```
22:00:16      SELECT nome FROM Fornecedor INNER JOIN  
Componente on Fornecedor.cnpj = Componente.Fornecedor_cnpj  
WHERE cidade = 'Belo Horizonte' LIMIT 0, 1000      37 row(s)  
returned      0.187 sec / 0.000 sec
```

4. Obter os clientes do estado de Santa Catarina(exemplo) que são atendidos pela transportadora X.

```
SELECT nome, cpf  
FROM Cliente, Transportadora  
WHERE Cliente.Transportadora_cnpj = Transportadora.cnpj  
AND Cliente.estado = "Santa Catarina"  
AND Transportadora.cnpj = "13321341000131"
```

```
22:02:30      SELECT nome, cpf FROM Cliente, Transportadora  
WHERE Cliente.Transportadora_cnpj = Transportadora.cnpj AND  
Cliente.estado = 'Belo Horizonte' AND Transportadora.cnpj =  
'1234567' LIMIT 0, 1000    0 row(s) returned    0.125 sec / 0.000 sec
```

5. Obter o extrato dos componentes requisitados por clientes do estado da Paraíba(exemplo).

```
SELECT idComponente, marca, nome, Fornecedor_cnpj
FROM Cliente, Componente, Cliente_Componente
WHERE Cliente.cpf = Cliente_Componente.Cliente_cpf
AND Componente.idComponente =
Cliente_Componente.Componente_idComponente
AND Cliente.estado = "Paraíba"
```

```
22:05:37      SELECT idComponente, marca, Cliente.nome,
Fornecedor_cnpj FROM Cliente, Componente, Cliente_Componente
WHERE Cliente.cpf = Cliente_Componente.Cliente_cpf AND
Componente.idComponente =
Cliente_Componente.Componente_idComponente AND
Cliente.estado = 'Belo Horizonte' LIMIT 0, 1000 14 row(s) returned
0.016 sec / 0.000 sec
```

6. Obter o CPF dos operadores que irão transportar o componente 52789063 para clientes do Estado de Minas Gerais(exemplo)

```
SELECT CPF
FROM ((Componente_Transportadora INNER JOIN Operador
      on Componente_Transportadora.Operador_cpf = Operador.cpf)
      NATURAL JOIN Cliente)
WHERE estado = "Minas Gerais" AND
      Componente_idComponente = "52789063"
```

```
22:07:58      SELECT CPF FROM ((Componente_Transportadora
INNER JOIN Operador on Componente_Transportadora.Operador_cpf
= Operador.cpf) NATURAL JOIN Cliente) WHERE estado = 'Minas
Gerais' AND Componente_idComponente = '52789063' LIMIT 0, 1000
0 row(s) returned    0.062 sec / 0.000 sec
```

7. Obter o CNPJ dos fornecedores para clientes da cidade de Brasília(exemplo)

```
SELECT Fornecedor_cnpj FROM
Componente NATURAL JOIN
(SELECT Componente_idComponente as idComponente FROM
  Cliente_Componente NATURAL JOIN
  (SELECT cpf as Cliente_cpf FROM Cliente WHERE cidade =
```

'Brasilia'))

8. Obter o CNPJ das transportadoras que carregam o componente Y para clientes do Estado do RJ(exemplo)

```
SELECT cnpf FROM  
(fornecedor )
```

9. Contar as peças que o cliente X requisitou ao fornecedor Y

```
SELECT COUNT(idComponente) FROM  
(Componente NATURAL JOIN  
(SELECT Componente_idComponente as idComponente FROM  
  Cliente_Componente NATURAL JOIN  
(SELECT cpf as Cliente_cpf FROM Cliente WHERE nome =  
  'X')))) WHERE Fornecedor_cnpj = 'Y'
```

10. Obter o total de SSDs(exemplo de categoria) oferecidos pelo fornecedor X.

```
SELECT COUNT(idComponente) FROM Categoria NATURAL JOIN  
(SELECT Categoria_idCategoria as idCategoria, idComponente FROM  
  Componente NATURAL JOIN (SELECT cnpj as Fornecedor_cnpj  
    FROM Fornecedor WHERE cnpj='X')) WHERE Categoria.nome =  
'SSD'
```

5. Auto Avaliação

Renato: colaborou conjuntamente com os colegas na elaboração da proposta de como seriam os atributos, entidades e relacionamento entre eles, colaborou também com o descritivo do sistema na primeira entrega, colaborou com a criação do diagrama Entidade-Relacionamento do sistema em software de modelagem e também com a formulação de determinadas consultas no banco de dados.

Bruno: Colaborou na elaboração do diagrama de entidade-relacionamento, fornecendo ideias iniciais, atuou também na geração de dados para testes, e em outros auxílios organizacionais.

Bernardo: Responsável por traduzir o esquema ER para o projeto lógico e participou da formulação das consultas.

Pedro: Responsável pela construção do diagrama UML e pela estruturação do Banco de Dados, a partir do *software* MySQL Workbench. Além disso, atuou junto com o grupo para a correção de erros durante a fase de popular o BD.

Colaborou para a construção de consultas.

Alex: Colaboração conjuntamente com os colegas na elaboração da proposta, na consultas SQL, popular o banco de dados e testes, além de resumir e apresentar o trabalho no relatório e em na elaboração da apresentação.

Vídeo de apresentação do trabalho:

<https://youtu.be/tqGUGIGFQI0>