

EP2 - Banco de Dados I

Grupo:

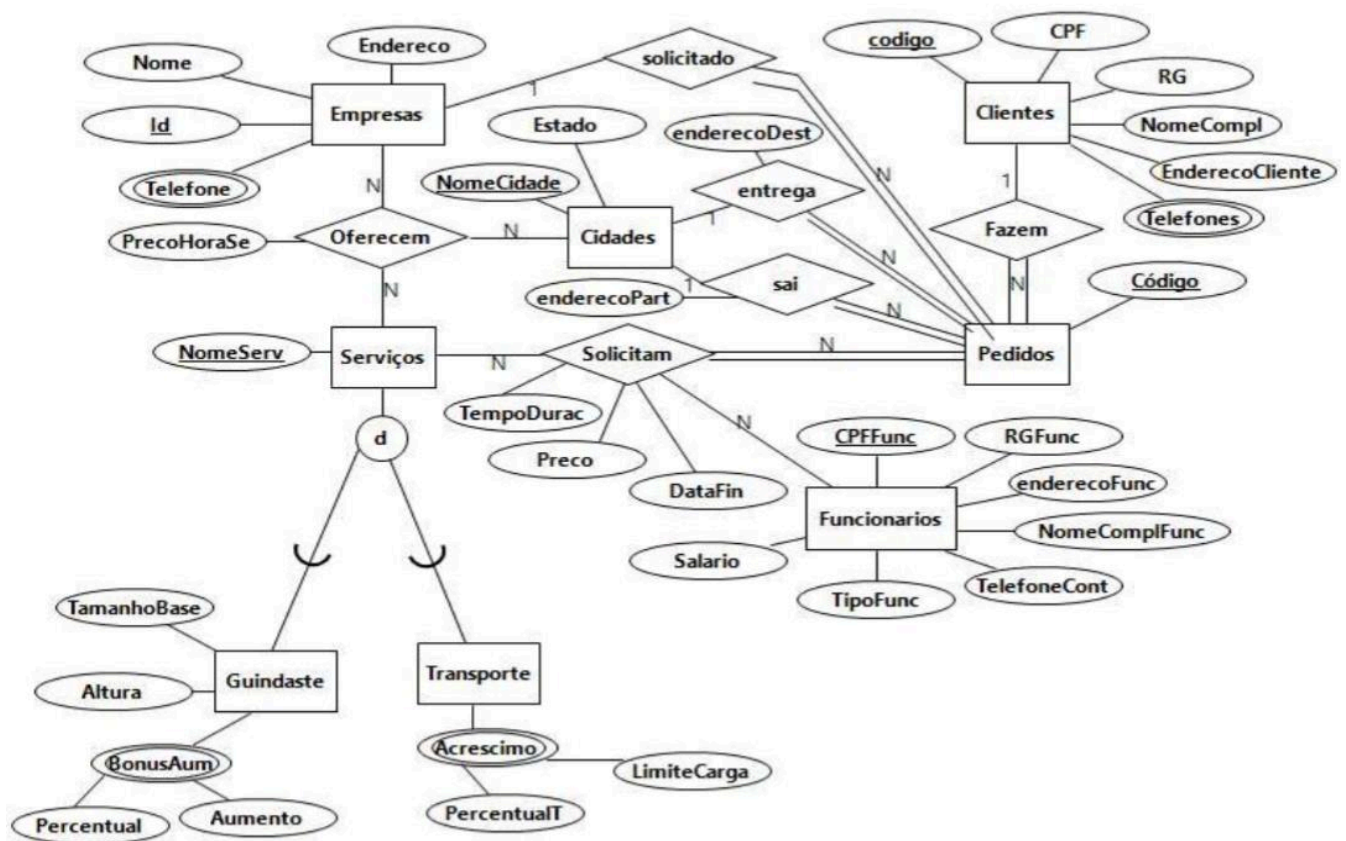
Aaron Ferraz - 14554474

Alaina R. dos Santos - 11837352

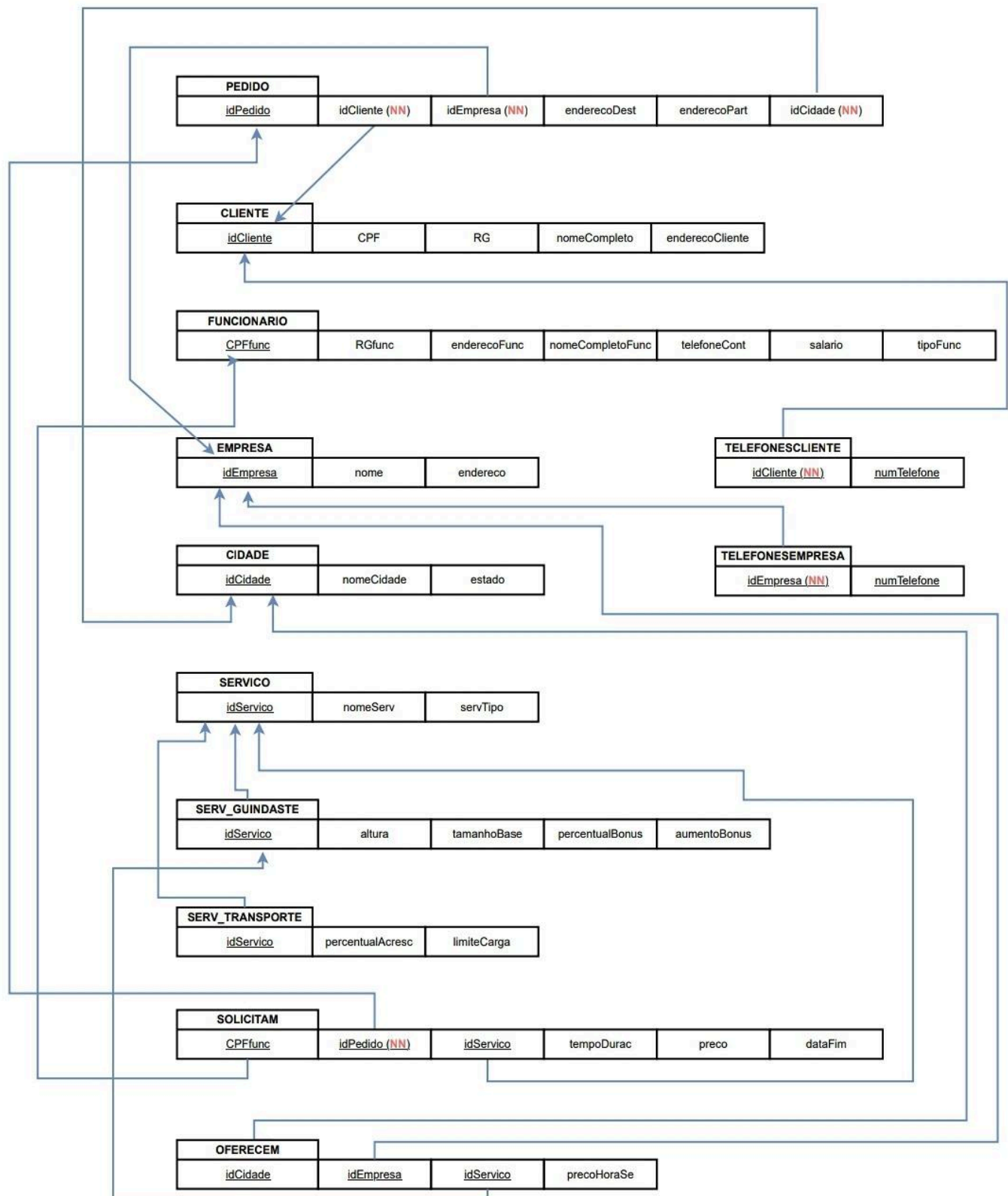
Lucas Gurgel - 14760234

Pedro Matos - 13684915

Modelo Entidade Relacionamento



Modelo Relacional



Criação das Tabelas em SQL

```
CREATE TABLE Empresa(  
    idEmpresa INTEGER NOT NULL,  
    nome VARCHAR(20) NOT NULL,  
    endereco VARCHAR(30),  
    CONSTRAINT ID_Empresa_PK PRIMARY KEY (idEmpresa)  
);
```

```
CREATE TABLE Servico(  
    idServico INTEGER NOT NULL,  
    nomeServ VARCHAR(40) UNIQUE NOT NULL,  
    servTipo VARCHAR(20) NOT NULL CHECK (servTipo IN ('Guindaste',  
    'Transporte')),  
    CONSTRAINT ID_Servico_PK PRIMARY KEY (idServico)  
);
```

```
CREATE TABLE Serv_guindaste (  
    idServico INT PRIMARY KEY,  
    altura DECIMAL(5, 2),  
    tamanhoBase DECIMAL(6, 2),  
    percentualBonus DECIMAL(3, 2),  
    aumentoBonus DECIMAL(7, 2),  
    CONSTRAINT FK_Guindaste_Servico FOREIGN KEY (idServico) REFERENCES  
    Servico(idServico) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE Serv_transporte (  
    idServico INT PRIMARY KEY,  
    percentualAcres DECIMAL(5, 2),  
    limiteCarga DECIMAL(7, 2),  
    CONSTRAINT FK_Transporte_Servico FOREIGN KEY (idServico) REFERENCES  
    Servico(idServico) ON DELETE CASCADE ON UPDATE CASCADE
```

);

CREATE TABLE Pedido (

idPedido INTEGER NOT NULL,

idCliente INT NOT NULL,

idEmpresa INT NOT NULL,

idCidade INT NOT NULL,

enderecoDest VARCHAR(150),

enderecoPart VARCHAR(150),

CONSTRAINT ID_Pedido_PK PRIMARY KEY (idPedido),

CONSTRAINT FK_Pedido_Cliente FOREIGN KEY (idCliente) REFERENCES
Cliente(idCliente) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK_Pedido_Empresa FOREIGN KEY (idEmpresa) REFERENCES
Empresa(idEmpresa) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK_Pedido_Cidade FOREIGN KEY (idCidade) REFERENCES
Cidade(idCidade) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Solicitam (

idServico INTEGER,

idPedido INTEGER NOT NULL,

CPFfunc CHAR(11),

tempoDurac DECIMAL(5, 2) NOT NULL,

preco DECIMAL(10, 2) NOT NULL CHECK(preco > 0),

dataFim DATE NOT NULL,

CONSTRAINT FK_Solicitam_Servico FOREIGN KEY (idServico) REFERENCES
Servico(idServico) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK_Solicitam_Pedido FOREIGN KEY (idPedido) REFERENCES
Pedido(idPedido) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK_Solicitam_CPFfunc FOREIGN KEY (CPFfunc) REFERENCES
Funcionario(CPFfunc) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT Servico_Pedido_CPFfunc_PK PRIMARY KEY(idServico, idPedido,
CPFfunc)

);

```
CREATE TABLE Cliente (  
    idCliente INTEGER NOT NULL,  
    CPF CHAR(11) UNIQUE NOT NULL,  
    RG VARCHAR(12) UNIQUE NOT NULL,  
    nomeCompleto VARCHAR(100) NOT NULL,  
    endereco VARCHAR(50),  
    CONSTRAINT ID_Cliente_PK PRIMARY KEY (idCliente)
```

```
);
```

```
CREATE TABLE Funcionario (  
    CPFfunc CHAR(11) NOT NULL,  
    enderecoFunc VARCHAR(50),  
    RGfunc VARCHAR(20) UNIQUE NOT NULL,  
    salario DECIMAL(10, 2) NOT NULL,  
    tipoFunc VARCHAR(50),  
    telefoneCont VARCHAR(15),  
    nomeCompletoFunc VARCHAR(100) NOT NULL,  
    CONSTRAINT ID_CPFfunc_PK PRIMARY KEY (CPFfunc)
```

```
);
```

```
CREATE TABLE Cidade (  
    idCidade INTEGER NOT NULL,  
    nomeCidade VARCHAR(40) UNIQUE NOT NULL,  
    estado VARCHAR(2),  
    CONSTRAINT ID_Cidade_PK PRIMARY KEY (idCidade)
```

```
);
```

```
CREATE TABLE TelefonesCliente (  
    idCliente INT NOT NULL,  
    numTelefone VARCHAR(15) NOT NULL,  
    CONSTRAINT Cliente_numTelefone_PK PRIMARY KEY(idCliente, numTelefone),  
    CONSTRAINT FK_TelefonesCliente_Cliente FOREIGN KEY (idCliente)
```

```

REFERENCES Cliente(idCliente) ON DELETE CASCADE ON UPDATE
CASCADE
);

CREATE TABLE TelefonesEmpresa (
    idEmpresa INT NOT NULL,
    numTelefone VARCHAR(15) NOT NULL,
    CONSTRAINT Empresa_numTelefone_PK PRIMARY KEY(idEmpresa,
numTelefone),
    CONSTRAINT FK_TelefonesEmpresa_Empresa FOREIGN KEY (idEmpresa)
REFERENCES Empresa(idEmpresa) ON DELETE CASCADE ON UPDATE
CASCADE
);

CREATE TABLE Oferecem (
    idEmpresa INT,
    idCidade INT,
    idServico INT,
    precoHoraSe DECIMAL(6,2) NOT NULL CHECK(precoHoraSe > 0),
    CONSTRAINT FK_Oferecem_Empresa FOREIGN KEY (idEmpresa)
REFERENCES Empresa(idEmpresa) ON DELETE CASCADE ON UPDATE
CASCADE,
    CONSTRAINT FK_Oferecem_Cidade FOREIGN KEY (idCidade) REFERENCES
Cidade(idCidade) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_Oferecem_Servico FOREIGN KEY (idServico) REFERENCES
Servico(idServico) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT Empresa_Servico_Cidade_PK PRIMARY KEY (idEmpresa,
idServico, idCidade)
);

```

Considerações

O enunciado trouxe algumas lacunas que tivemos que tomar decisões em cima do modelo relacional e criação de tabelas. Portanto, vamos evidenciá-las para esclarecer nosso ponto de vista.

Em relação às restrições, as especificações no enunciado não são claras. Dessa forma, decidimos considerar algumas coisas na criação das tabelas. Mesmo sem ter no modelo relacional, colocamos restrições em alguns atributos importantes para manter a consistência da tabela, como por exemplo um cliente sem nome não faria sentido ser armazenado. Além disso, foram determinados *id* para todas as tabelas, em que, por sua vez, suas chaves candidatas são estabelecidas como *UNIQUE* e *NOT NULL*, pois a cláusula *UNIQUE* permite valores nulos.

As chaves estrangeiras têm nomes iguais às colunas de outras tabelas. Isso pode ser resolvido especificando a tabela de cada atributo. Entretanto, respeitamos a restrição de integridade que em uma tabela não pode haver mais de um atributo com o mesmo nome.

Resolvemos os telefones multivalorados de cliente e empresa com a criação das tabelas *TelefonesCliente* e *TelefonesEmpresa*. Também foi aplicado *NOT NULL* no *idCliente* para que um número esteja sempre associado a um cliente – não faria sentido ter um número sem um cliente.

Na criação da tabela "*Servico*", observou-se linha única na especialização, indicando que nem todo serviço é de transporte ou guindaste, e portanto o atributo *servTipo* para identificar o tipo do serviço. Contudo como no banco de dados não consideramos serviços além dos já especificados, criamos uma restrição que só aceita a inserção do serviço ser do tipo Guindaste ou Transporte, corroborando com o modelo relacional criado. Não seria coesa a existência de um serviço de outro tipo se não há uma entidade para representá-lo.

Em virtude de corresponder com a integridade referencial, colocamos o *CASCADE* tanto ao excluir quanto ao atualizar, isso significa que quando for feito uma dessas duas ações, todas as dependências daquela chave vão ser excluídas/atualizadas de forma automática – contribuindo com a consistência e integridade do banco de dados.

Atributos que apresentam (*NN*) após seus nomes são obrigatoriamente não-nulos pela restrição de integridade da participação total.

Durante a criação das tabelas em *SQL* foi pensado mais acerca de seus atributos e restrições, de tal modo que foram criadas sem ordem específica. Posto isso, se o código fosse implementado em um SGBD real, a ordem deveria ser pensada para evitar erros de referência, tendo tabelas com chaves referenciadas criadas antes de tabelas que as necessitam.

Por fim, com o intuito de trazer uma solução mais próxima das mostradas em aulas, usamos a sintaxe apropriada de nomear as restrições.