



Universidade Estadual da Paraíba

Banco de Dados

Projeto de Banco de Dados Relacional

Prof. Dr. Vladimir Costa Alencar

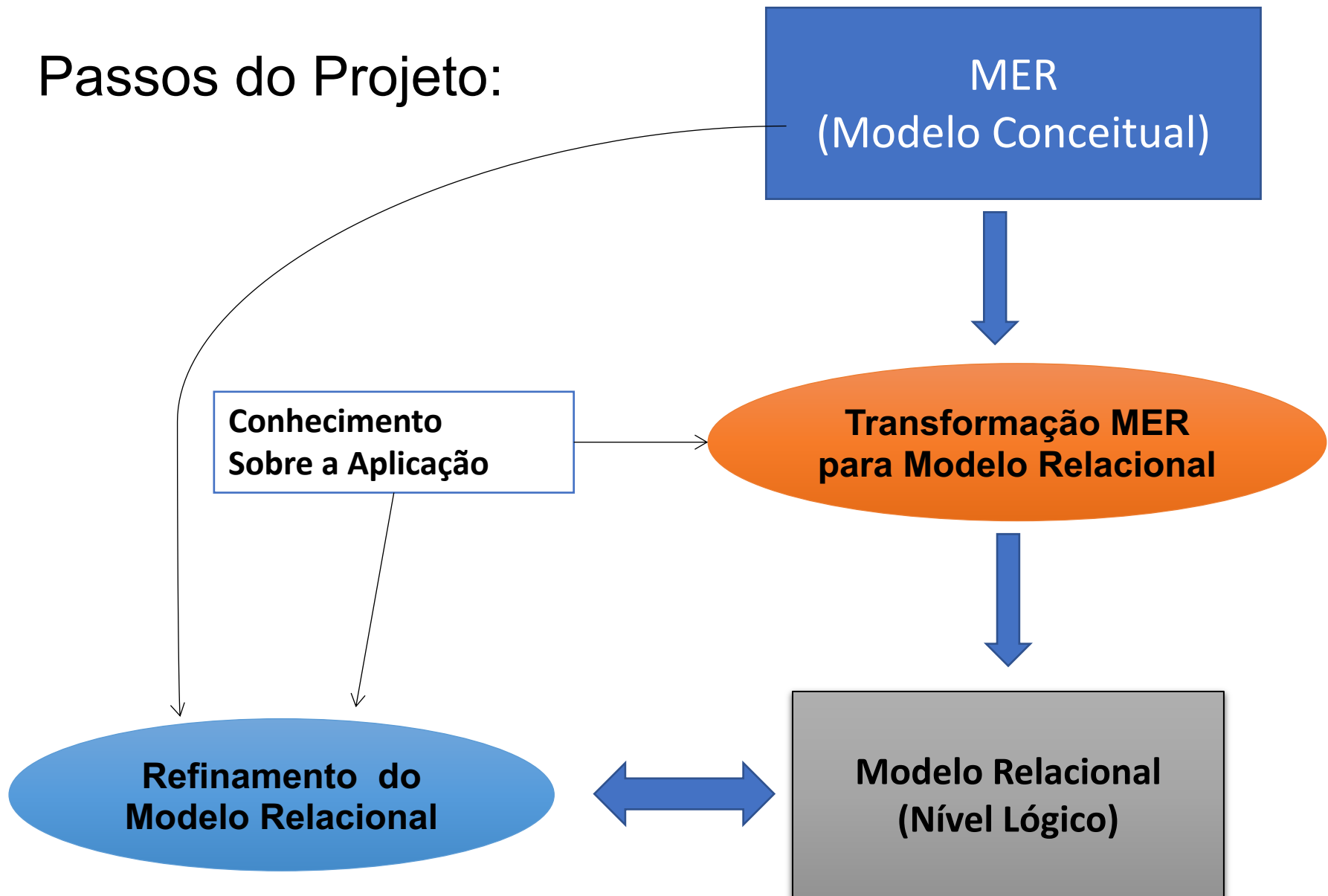
valencar @gmail.com

<https://www.valencar.com/>



Projeto de Banco de Dados Relacional

Passos do Projeto:



Projeto de Banco de Dados Relacional

Tem como objetivo:

- Gerar um conjunto de esquemas de relação (Tabelas)
- Existem regras para a transformação de um modelo MER para um modelo Relacional

Projeto de Banco de Dados Relacional

As regras foram definidas tendo em vista 2 objetivos básicos:

1. Obter um banco de dados que permita **boa performance** de instruções de consulta e alteração do banco de dados.
 - **Boa performance** significa basicamente diminuir o número de acessos a disco, já que estes consomem muito tempo na execução de uma instrução de banco de dados
2. Obter um banco de dados que **simplifique** o desenvolvimento e a manutenção de aplicações.

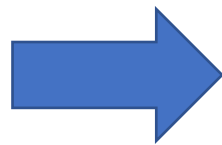
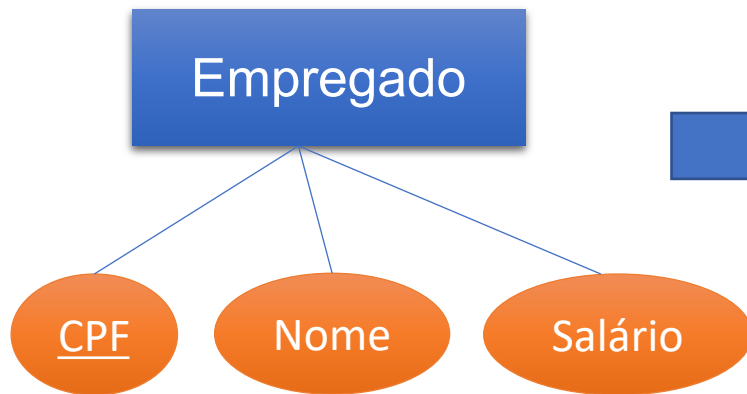
Principais Conceitos do MER (Modelo Entidade-Relacionamento)

- Tipos de entidades (regular, fraca);
- Graus de relacionamentos (binário, ternário, n-ário);
- Atributos (simples, compostos, multivalorados);
- Restrições (chaves, cardinalidade, etc);
- A seguir veremos regras para a tradução de MER para o esquema Relacional (tabelas)

Regra 1: Entidades Regulares

Para cada entidade regular E no esquema E-R, criamos uma relação R que inclui os atributos simples de E;

- Para cada atributo composto de E, incluimos somente os seus atributos simples;
- Escolhemos um dos atributos chaves de E para ser a chave primária de R.

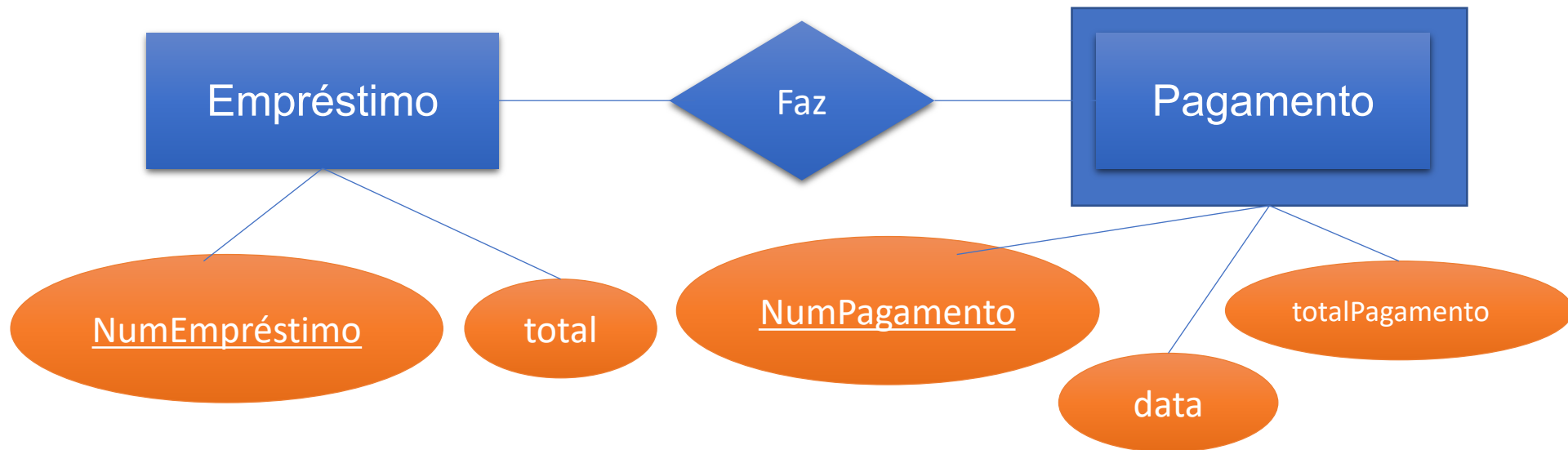


```
Create table Empregado(  
  CPF char(9),  
  Nome char(60),  
  Salario float  
  primary key (CPF)  
);
```

Empregado (CPF, Nome, Salário)

Regra 2: Entidades Fracas

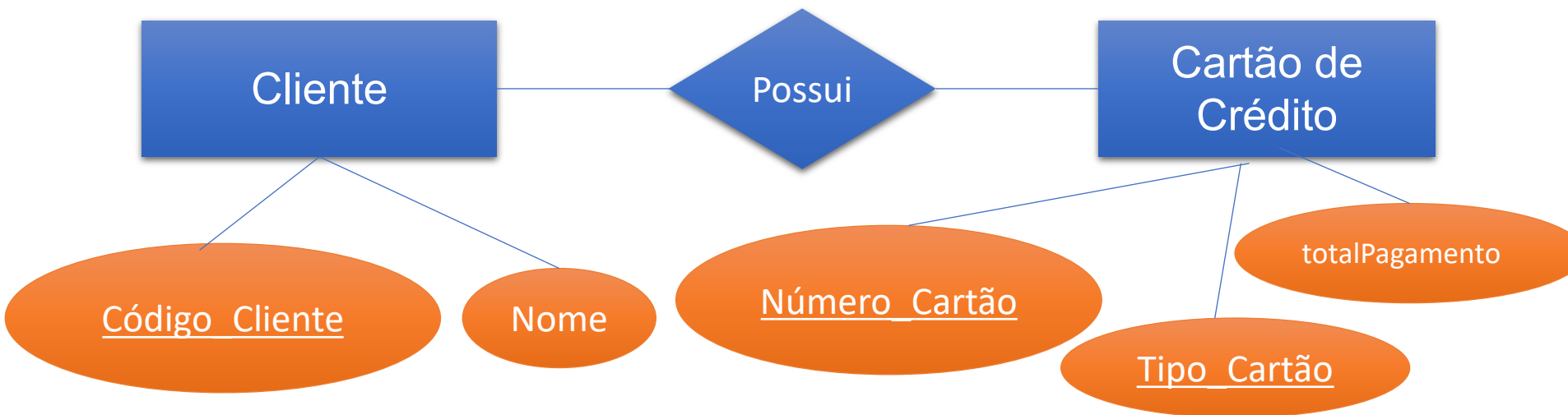
- Para cada entidade fraca W, com entidade forte E, no esquema MER, criamos uma relação R (tabela) e incluimos todos os atributos simples de W como atributos de R
- A chave primária de R é a combinação da chave primária da entidade forte E e a chave da entidade fraca W



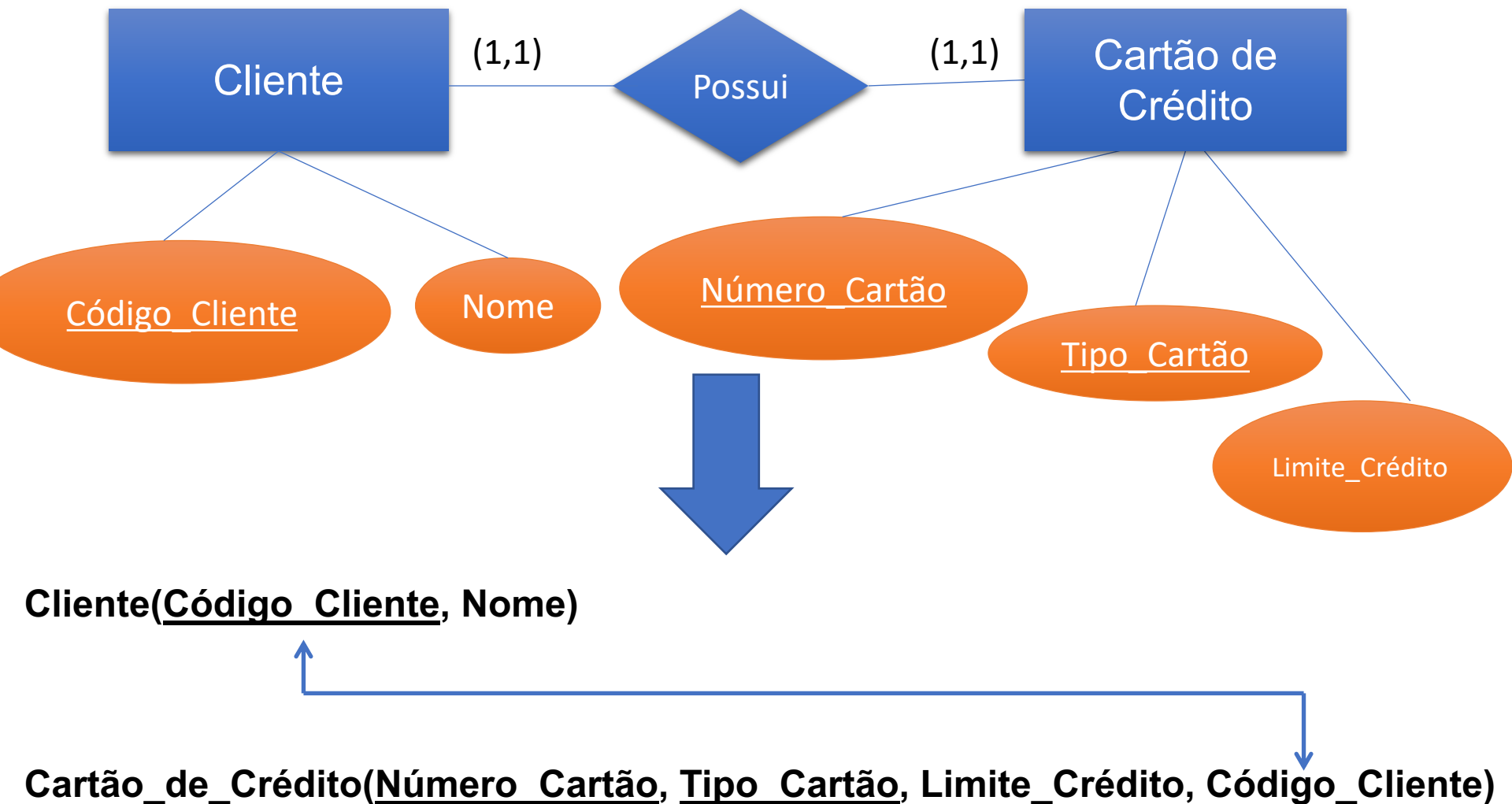
Empréstimo(NumEmpréstimo, NumPagamento, total, data, totalPagamento)

Regra 3: Relacionamentos 1:1

- Identificamos as relações S e T que correspondem às entidades que participam do relacionamento
- Escolhemos uma das relações, digamos S, e incluimos como chave estrangeira em S a chave primária de T
- É melhor escolher para desempenhar o papel de S a entidade que tenha participação total no relacionamento
- Incluimos todos os atributos simples do relacionamento 1:1 como atributos de S.

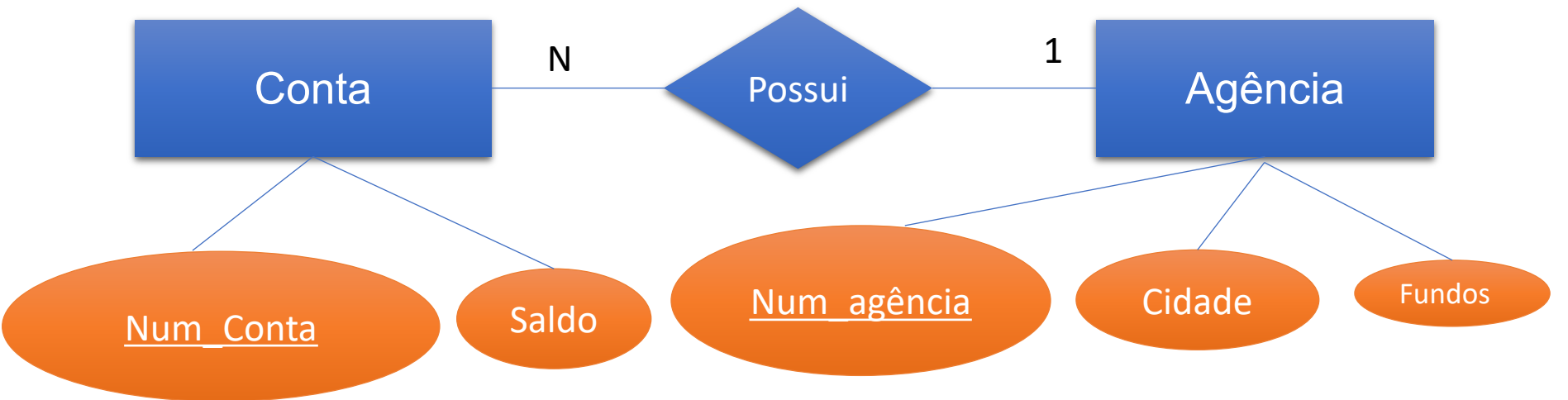


Regra 3: Relacionamentos 1:1



Regra 4: Relacionamentos 1:N

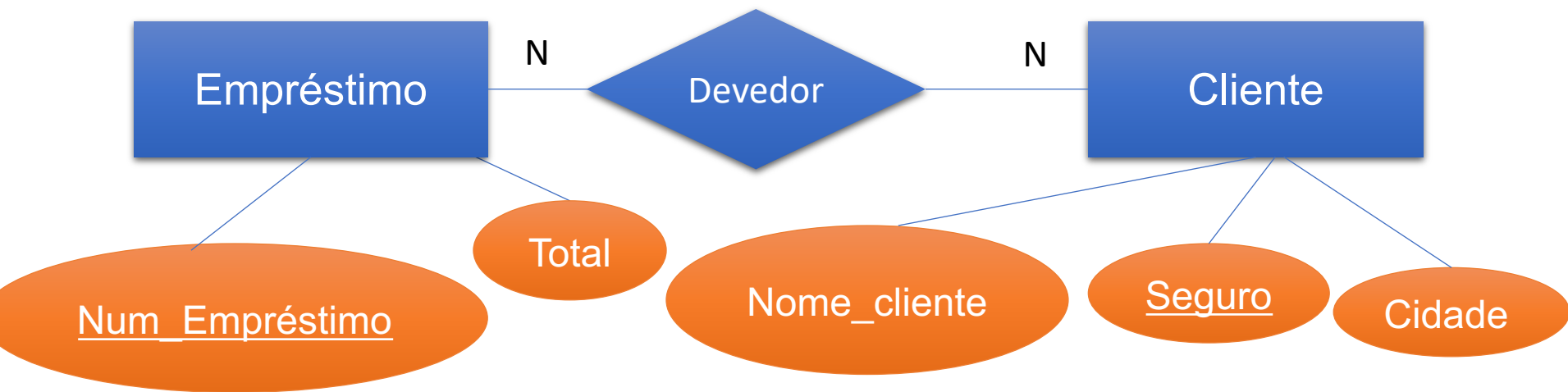
- Identificamos a relação S que representa a entidade que participa do lado N do relacionamento
- Incluímos como chave estrangeira em S a chave primária da relação T que representa a outra entidade (lado 1) que participa do relacionamento



Conta(Num_Conta, Saldo, Num_agência) ↔ Agência(Num_agência, Cidade, Fundos)

Regra 5: Relacionamentos N:M

- Criamos uma nova relação **S** para representar o relacionamento
- Incluímos como chave estrangeira em **S** as chaves primárias das relações que participam do relacionamento. A combinação destas chaves formará a chave primária da relação **S**.



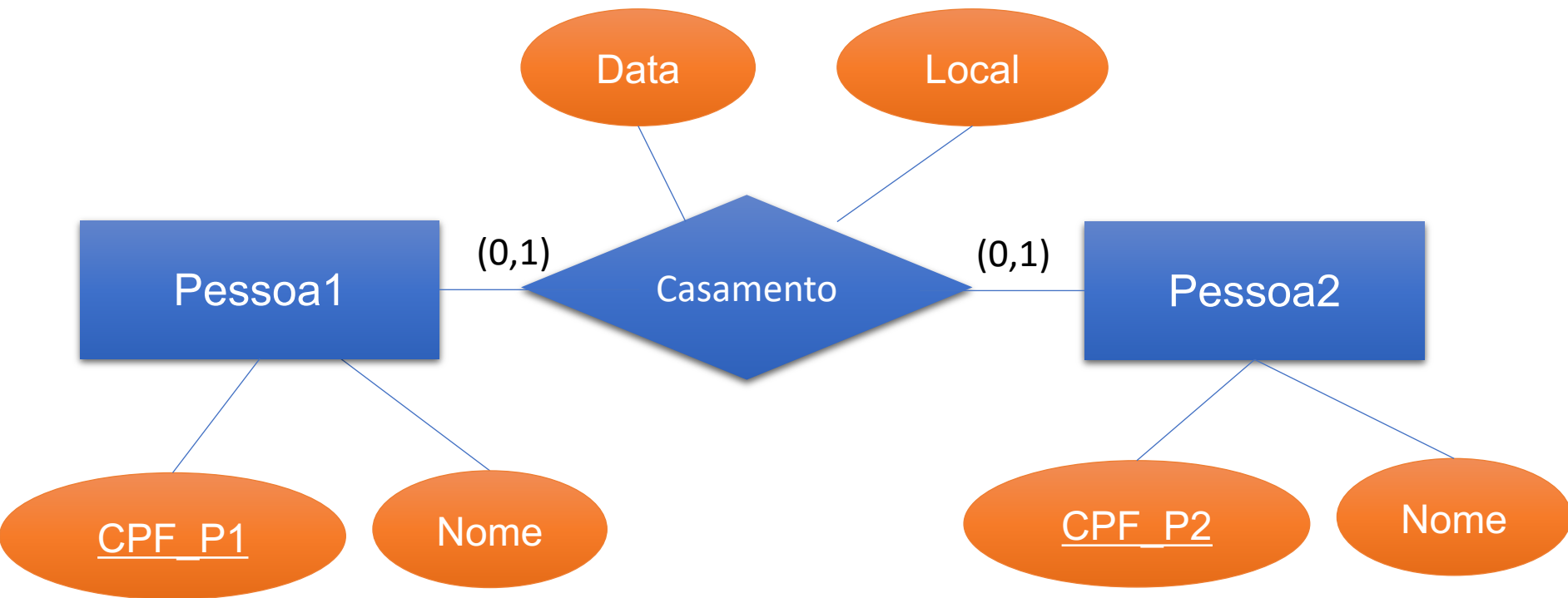
Empréstimo (Num_Empréstimo, total)

Cliente (Seguro, Nome_Cliente, Cidade)



Devedor (Num_Empréstimo, Seguro)

Ambas as entidades tem participação parcial



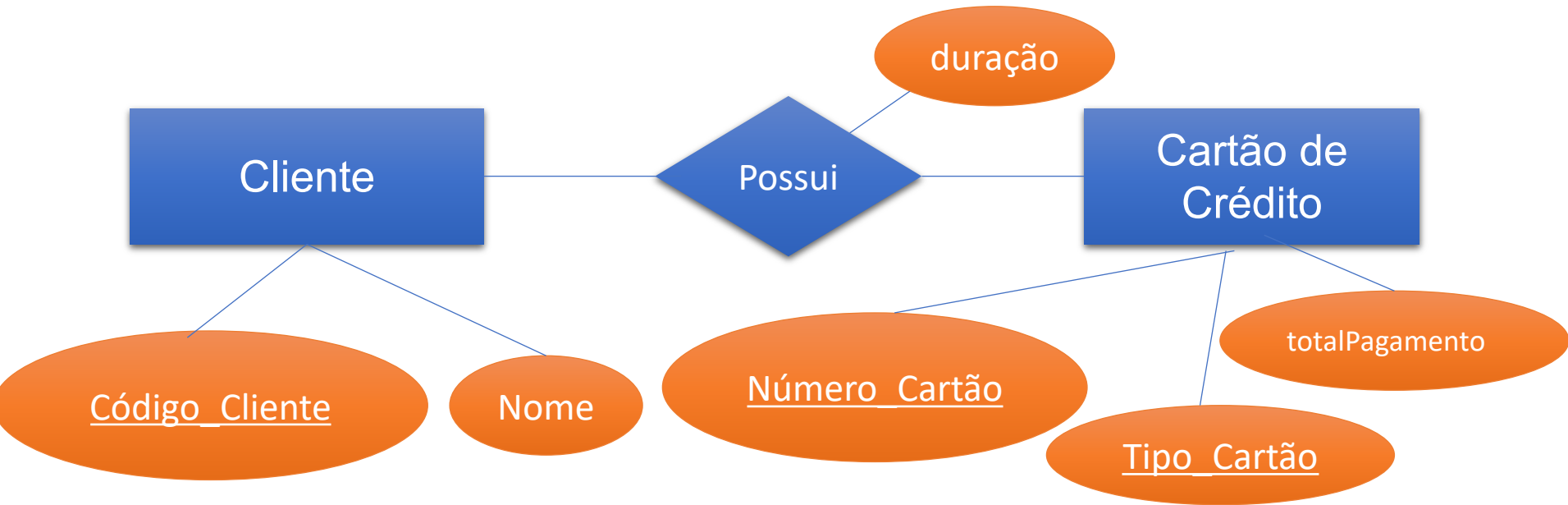
Pessoa1 (CPF_P1, Nome)

Pessoa2 (CPF_P2, Nome)



Casamento(CPF_P1, CPF_P2, Data, Local)

Relacionamento possui atributo



Cliente (Código_Cliente, Nome)

Cartão_de_Crédito (Número Cartão, Tipo Cartão, totalPagamento)

Possui (Código_Cliente, Número Cartão, Tipo Cartão, duração)

OU

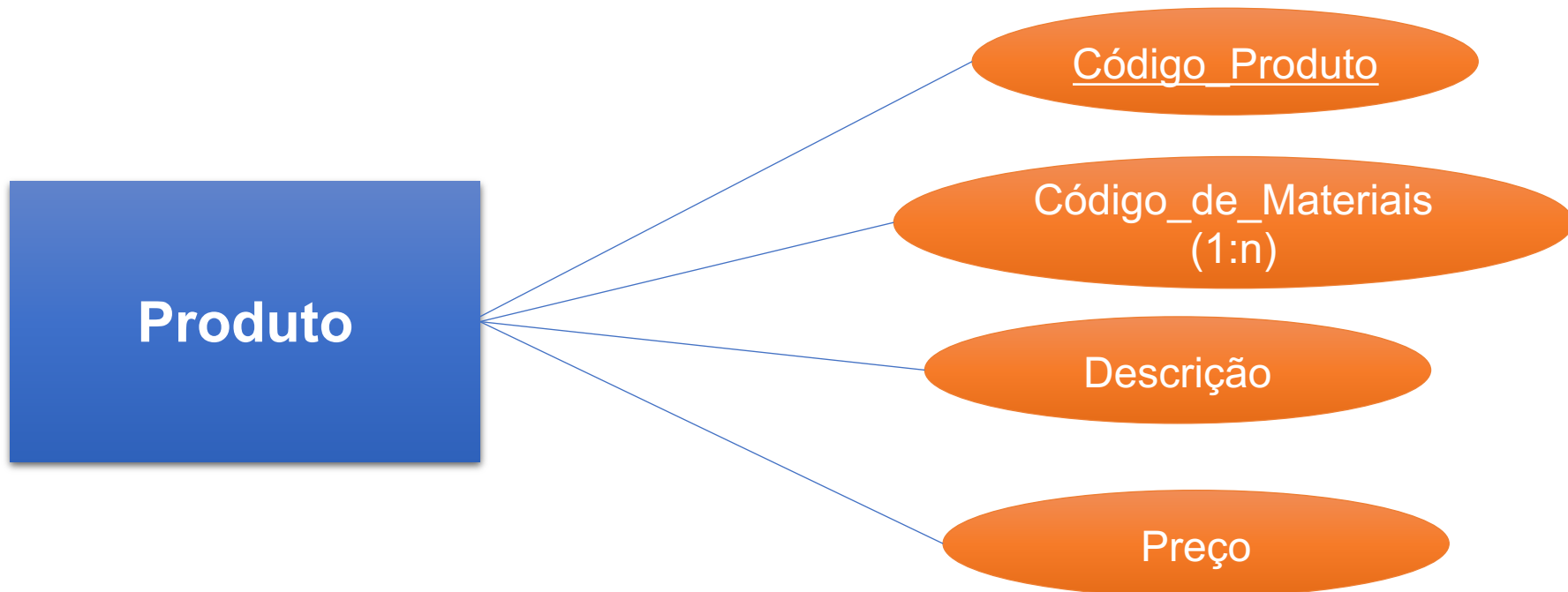
Cliente (Código_Cliente, Nome)

Cartão_de_Crédito (Número Cartão, Tipo Cartão, totalPagamento, Código_Cliente, duração)

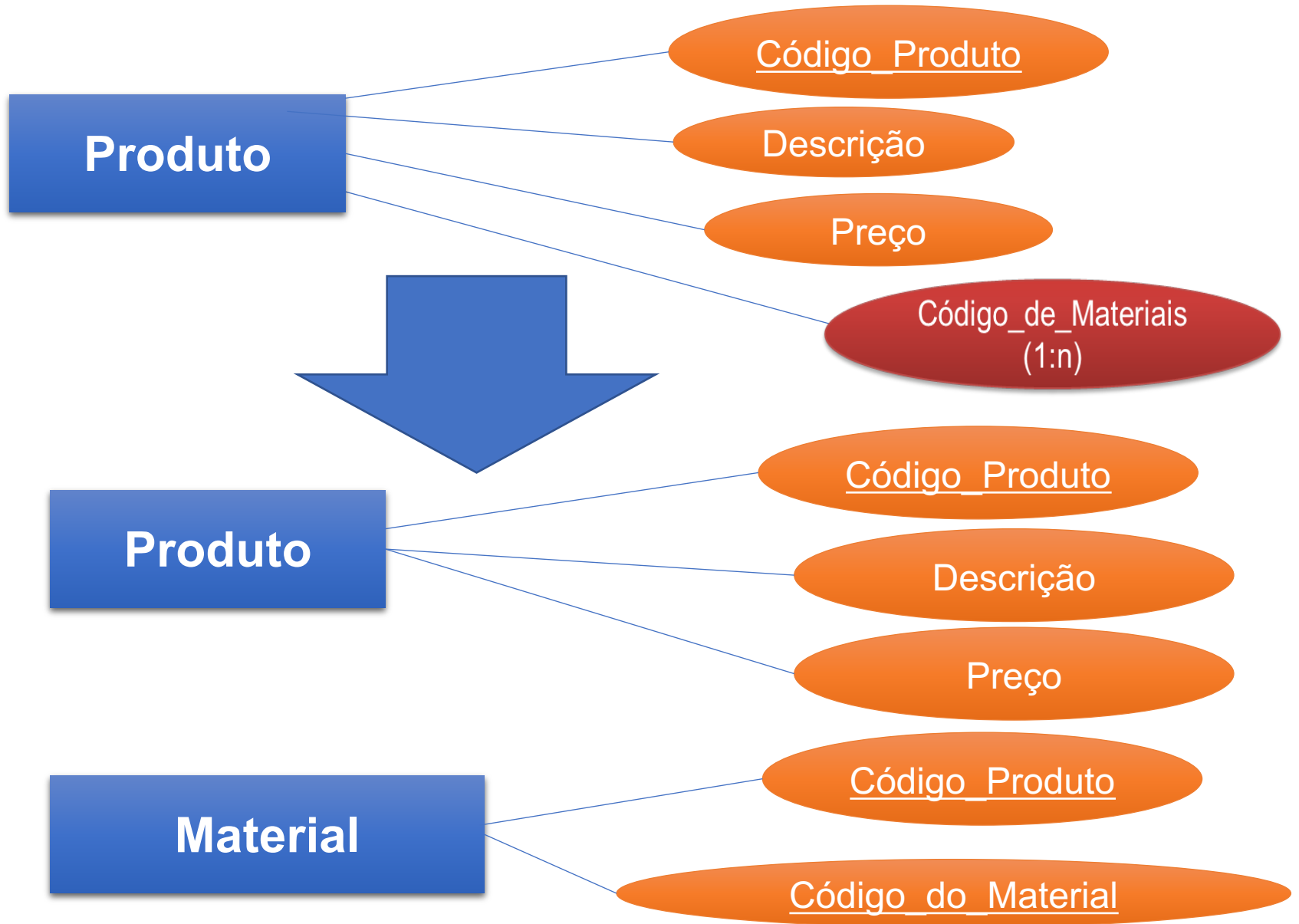


Regra 6: Atributos Multivalorados

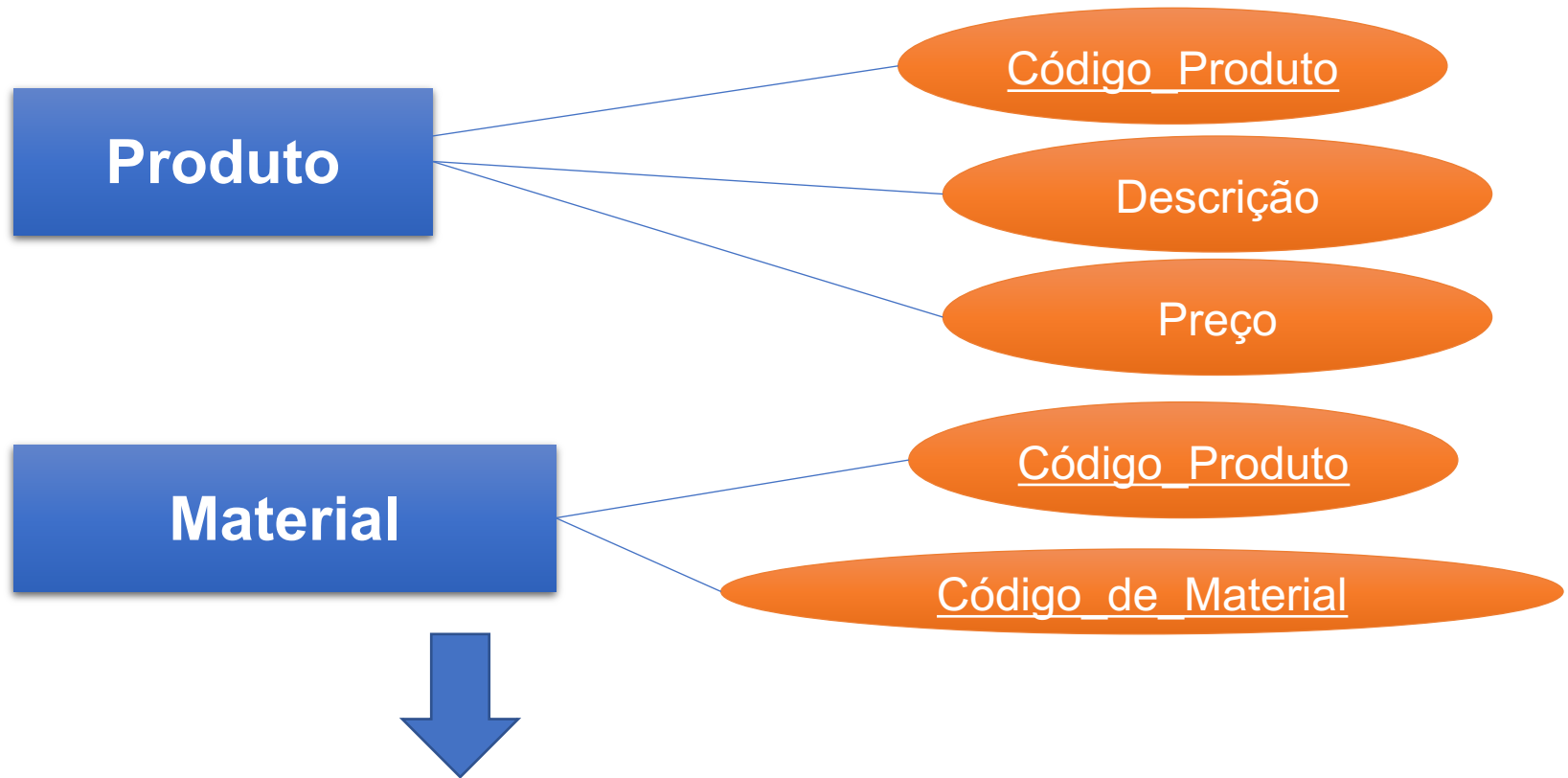
- Criamos uma nova relação **R** que inclui o atributo multivalorado **A** mais a chave primária **K** da relação que representa a entidade (ou relacionamento) que tem **A** como atributo;
- A chave primária de **R** é a combinação de **A** e **K**;



Regra 6: Atributos Multivalorados



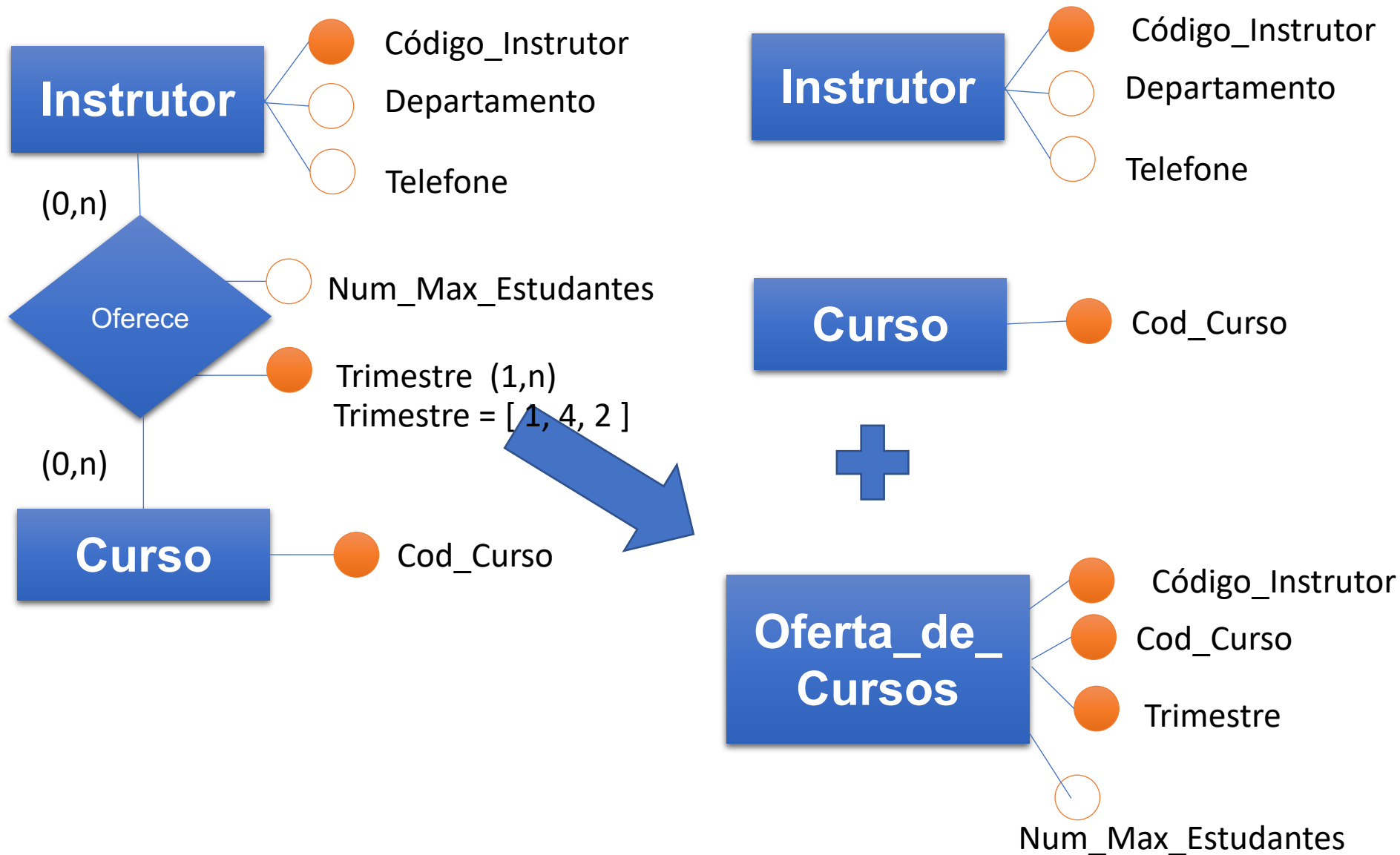
Regra 6: Atributos Multivalorados



Produto (Código_Produto, Descrição, Preço)

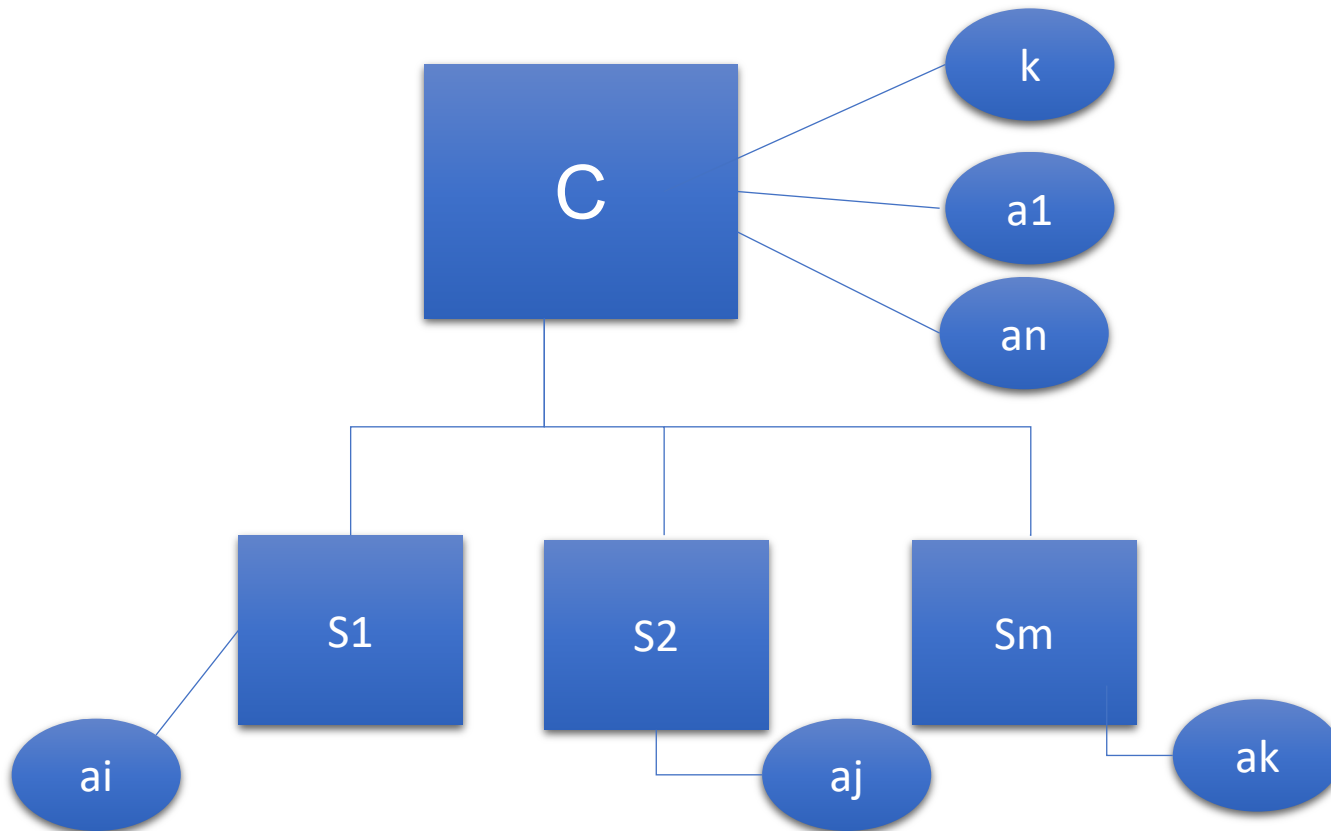
Produto_Material (Código_de_Material, Código_Produto,)

Regra 6: Atributos Multivalorados - Relacionamento



Regra 7: Especialização e Generalização

- Converta cada especialização com m subclasses $\{S1, S2, \dots, Sm\}$ e superclasse C , cujos atributos são $\{k, a1, \dots, an\}$ onde k é a chave primária, em esquemas de relações usando uma das seguintes opções:

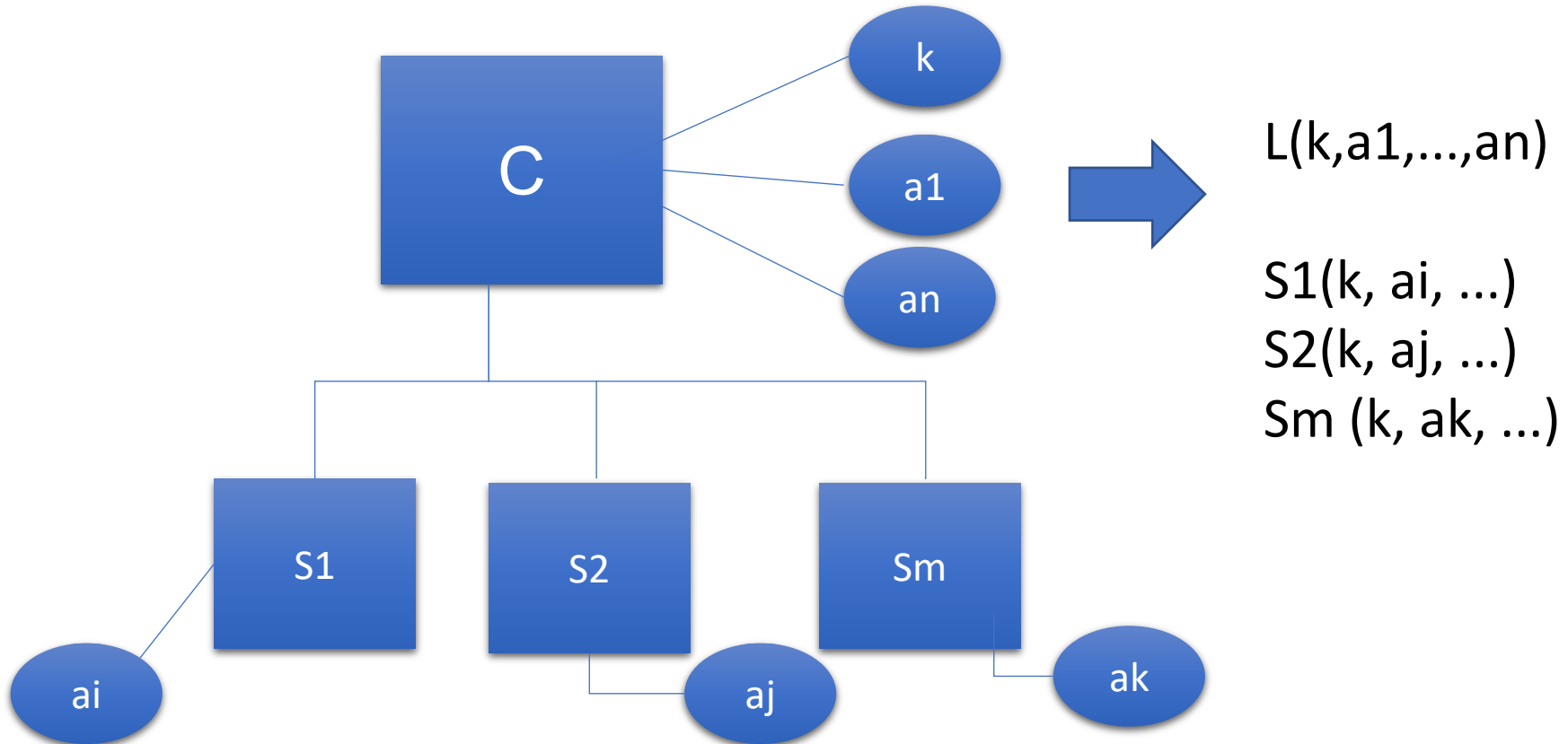


Regra 7: Especialização e Generalização

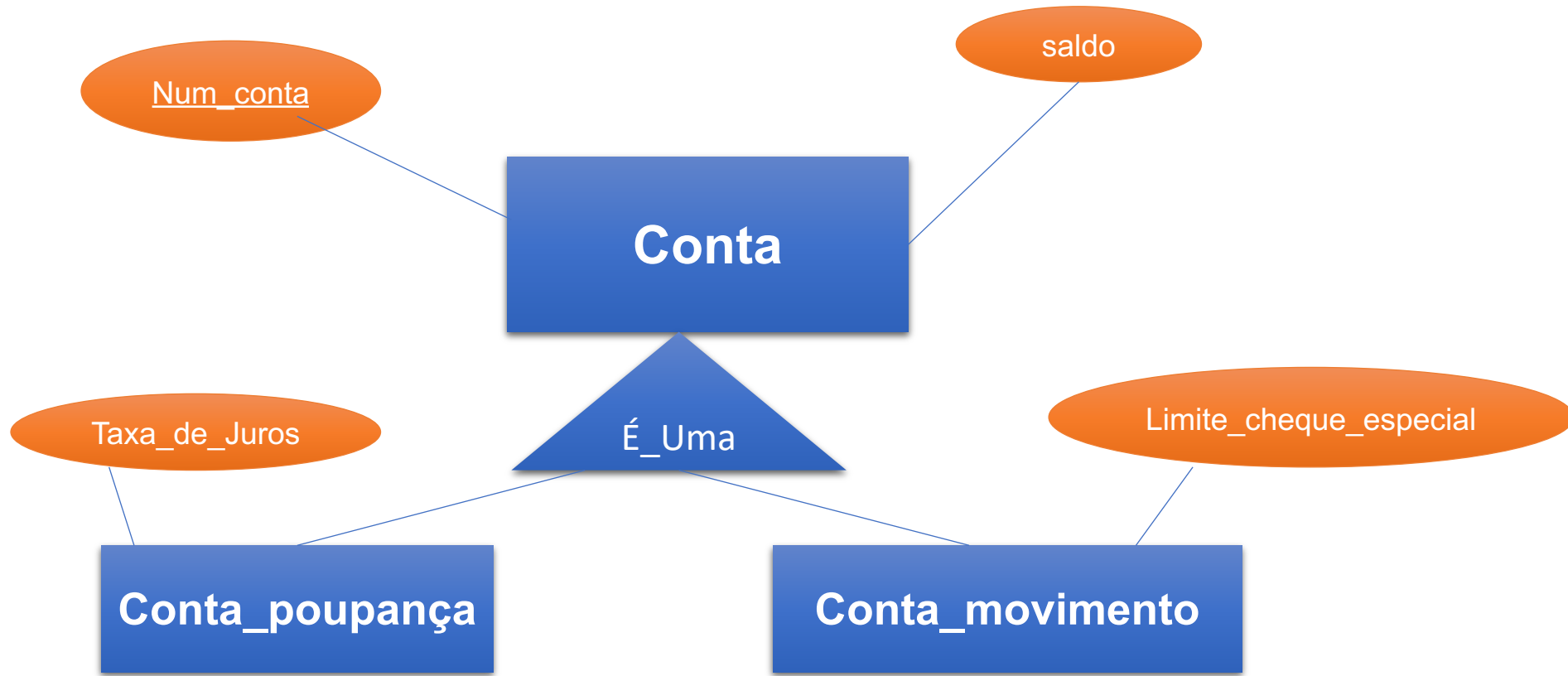
A) Criar uma Relação **L** para **C** com os atributos **Atrib(L) = {k, a1, ..., an}** e chave primária **k**.

Criar também uma Relação **Li** para cada subclasse **Si**, $1 \leq i \leq m$, com os seguintes atributos:

$\text{Atrib}(Li) = \{k\} \cup \{\text{atributos de } Si\}$, **k** será a chave primária.



Regra 7: Generalização e Especialização

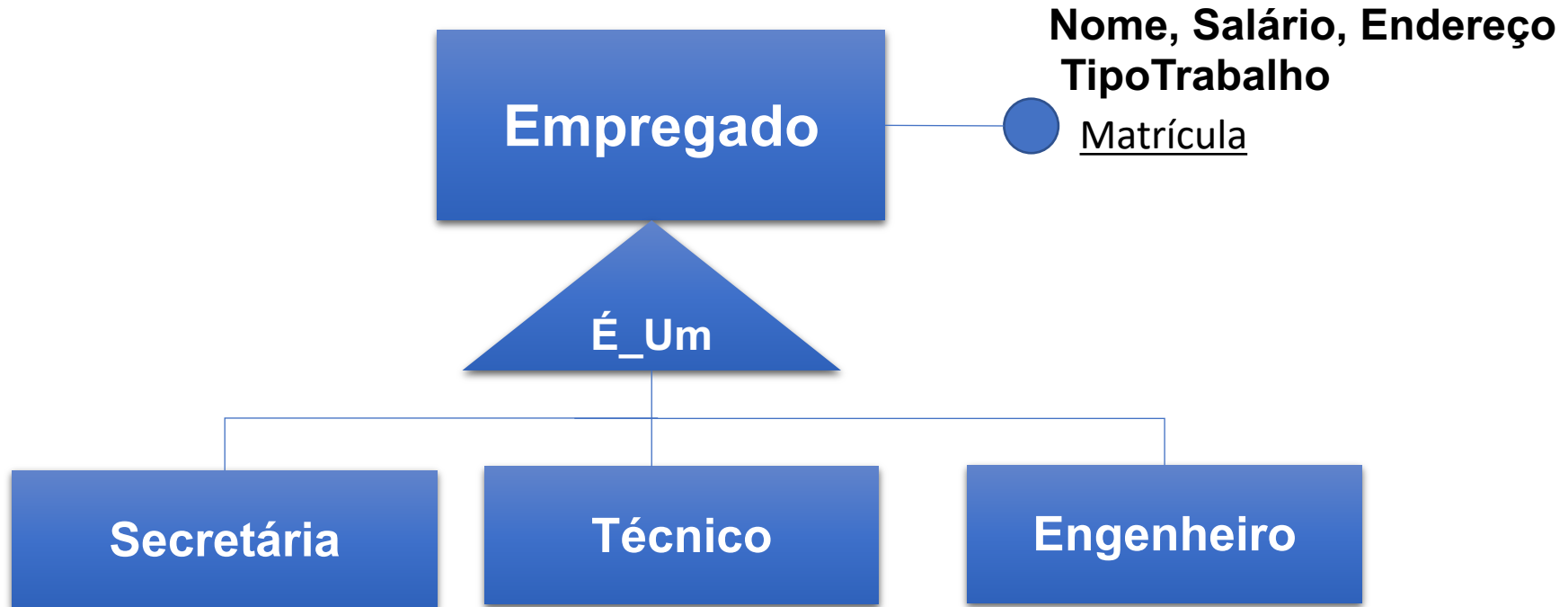


Conta(Num_conta, saldo)

Conta_poupança(Num_conta, Taxa_de_Juros)

Conta_movimento(Num_conta, Limite_cheque_especial)

Regra 7: Generalização e Especialização



Empregado(Matrícula, Nome, Salário, Endereço, TipoTrabalho)

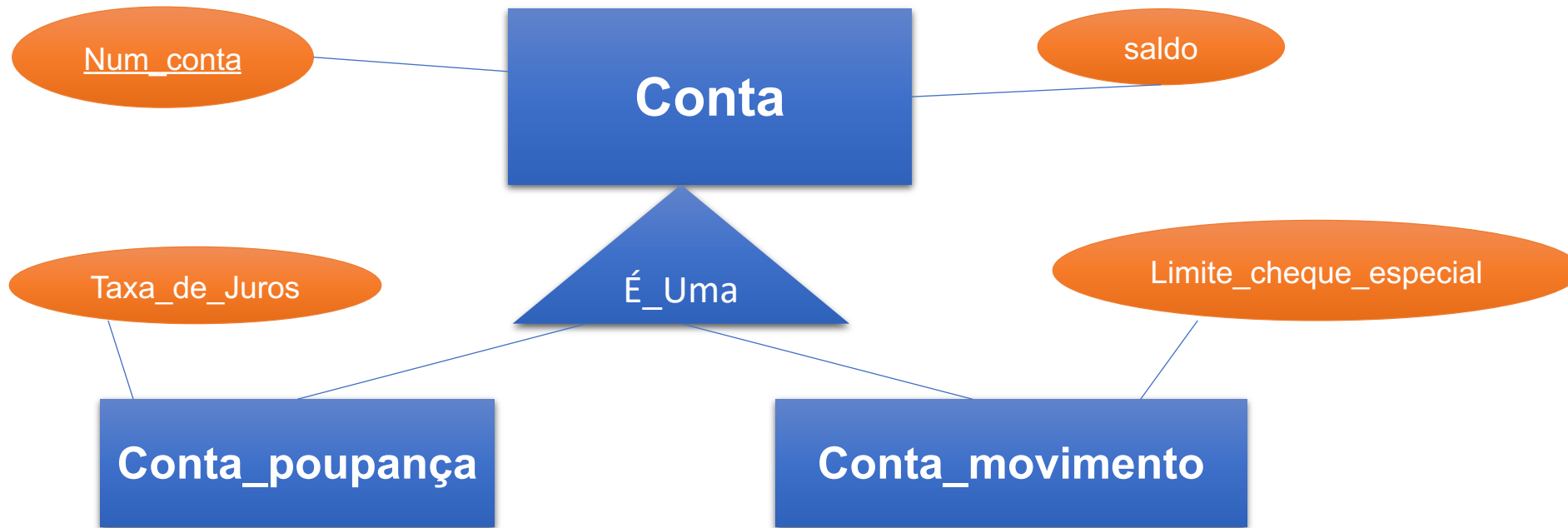
Secretária(Matrícula, VelocidadeDigitação)

Técnico(Matrícula, Especialidade)

Engenheiro(Matrícula, Especialidade, CREA)

Regra 7: Generalização e Especialização

B) Criar uma Relação L_i para cada **subclasse** S_i , $1 \leq i \leq m$, com os atributos : $\text{Atrib}(L_i) = \{\text{atributos de } S_i\} \cup \{k, a_1, \dots, a_n\}$ e chave primária $(L_i) = k$



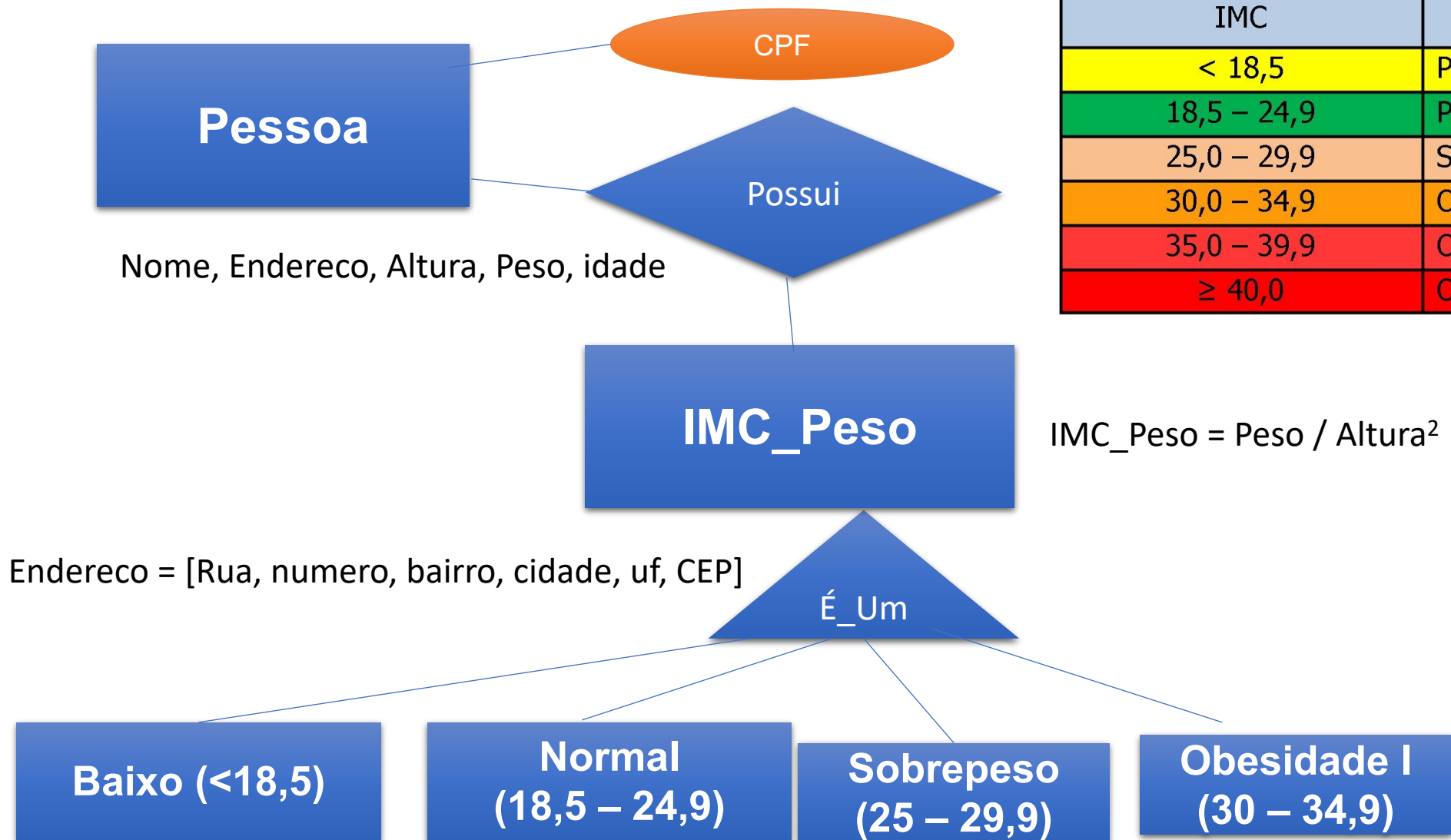
Conta_poupança(Num_conta, saldo, Taxa_de_Juros)

Conta_movimento(Num_conta, saldo, Limite_cheque_especial)

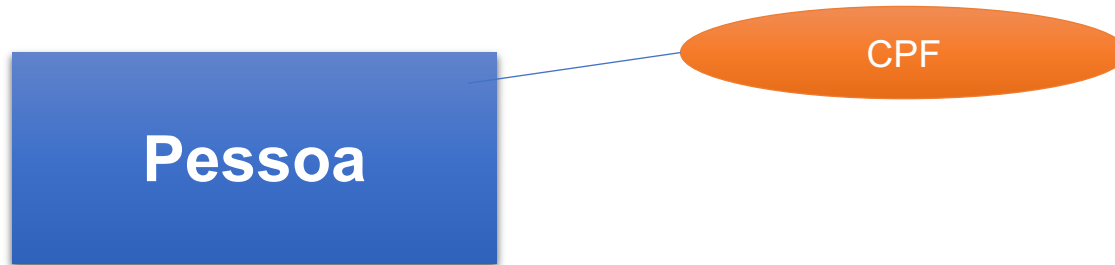
Regra 7: Generalização e Especialização



Regra 7: Generalização e Especialização



Regra 7: Generalização e Especialização



Nome, endereço, Altura, Peso, idade

OU

Pessoa(CPF, Nome, Rua, numero, bairro, cidade, uf, CEP, Altura, Peso, idade)

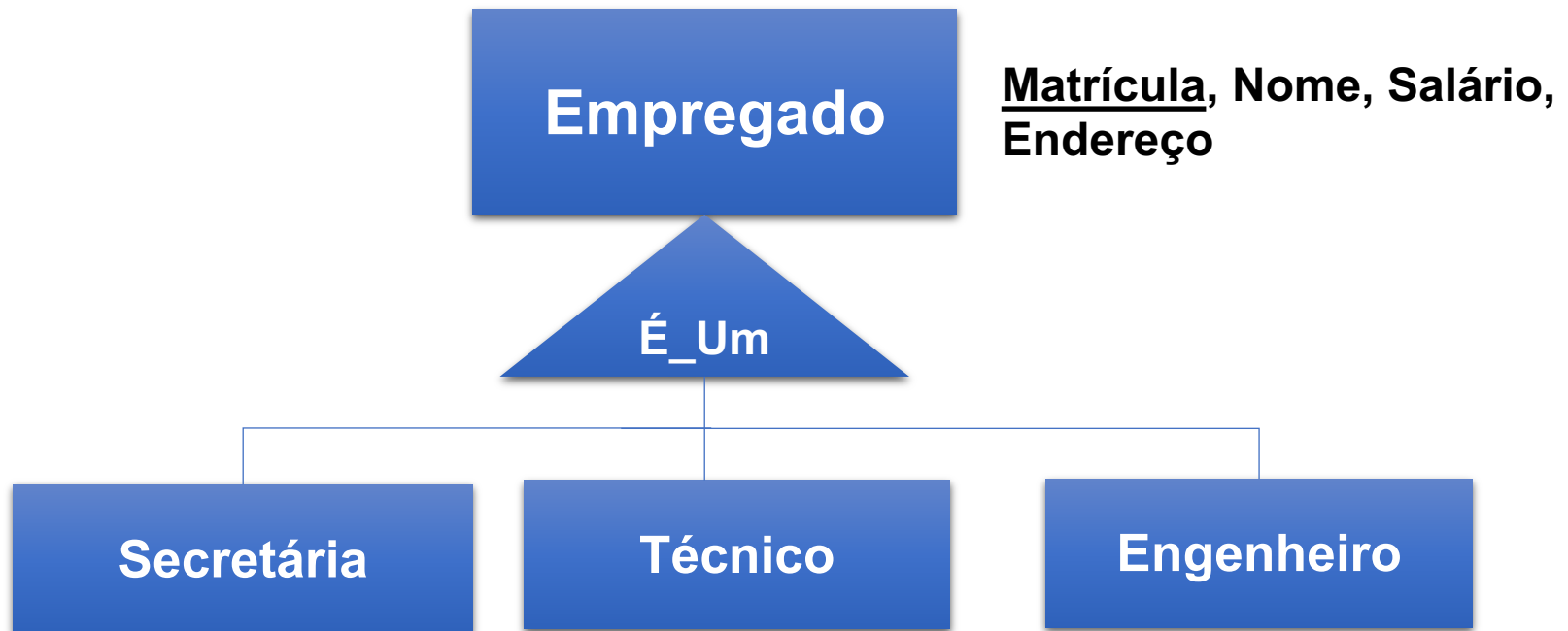
Endereço = [Rua, numero, bairro, cidade, uf, CEP] => Não é ATÔMICO (Vários atributos)

OU

Pessoa(CPF, Nome, codigo_endereco, Altura, Peso, idade)

Endereço (codigo_endereco, Rua, numero, bairro, cidade, uf, CEP)

Regra 7: Generalização e Especialização

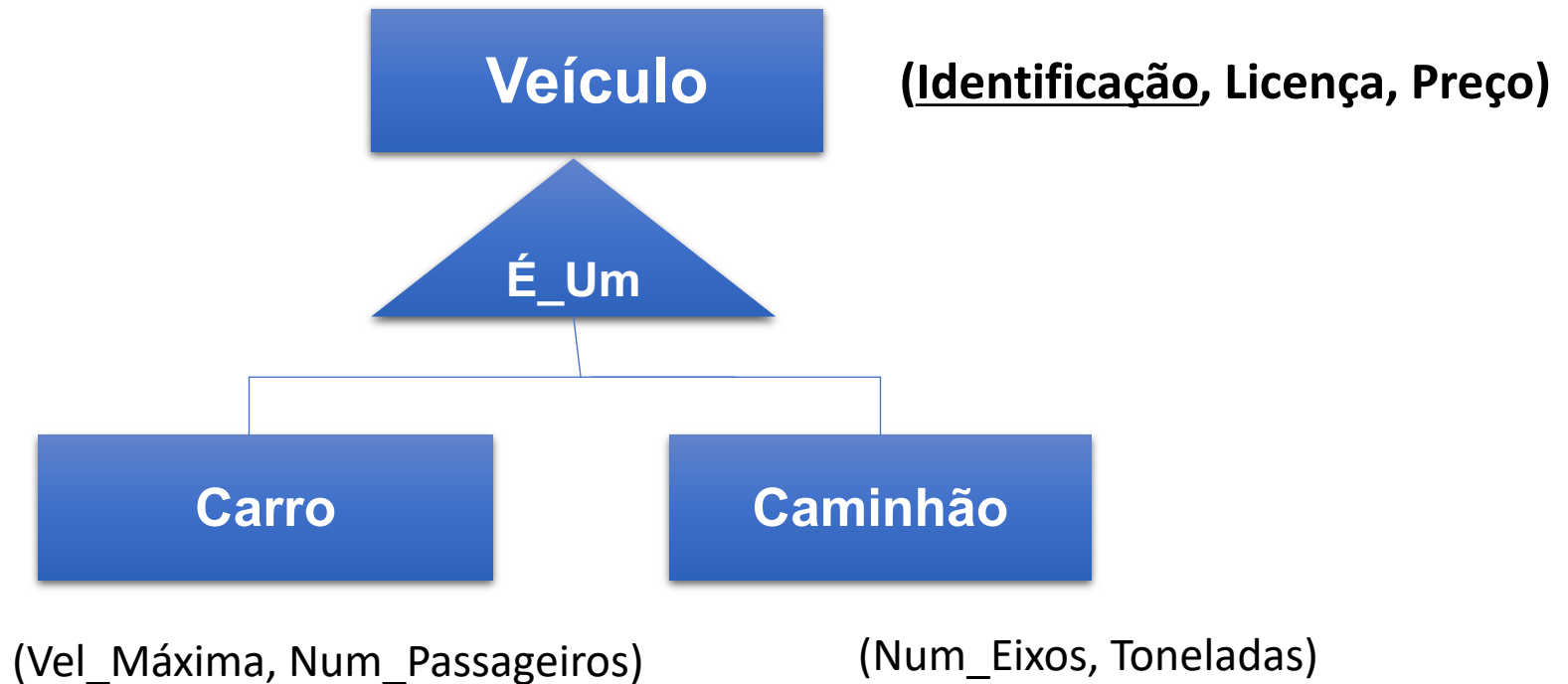


Secretária(Matrícula, Nome, Salário, Endereço, VelocidadeDigitação)

Técnico(Matrícula, Nome, Salário, Endereço, Especialidade)

Engenheiro(Matrícula, Nome, Salário, Endereço, Tipo_Engenheiro, CREA)

Regra 7: Generalização e Especialização



Carro(Identificação , Licença, Preço, Vel_Máxima, Num_Passageiros)

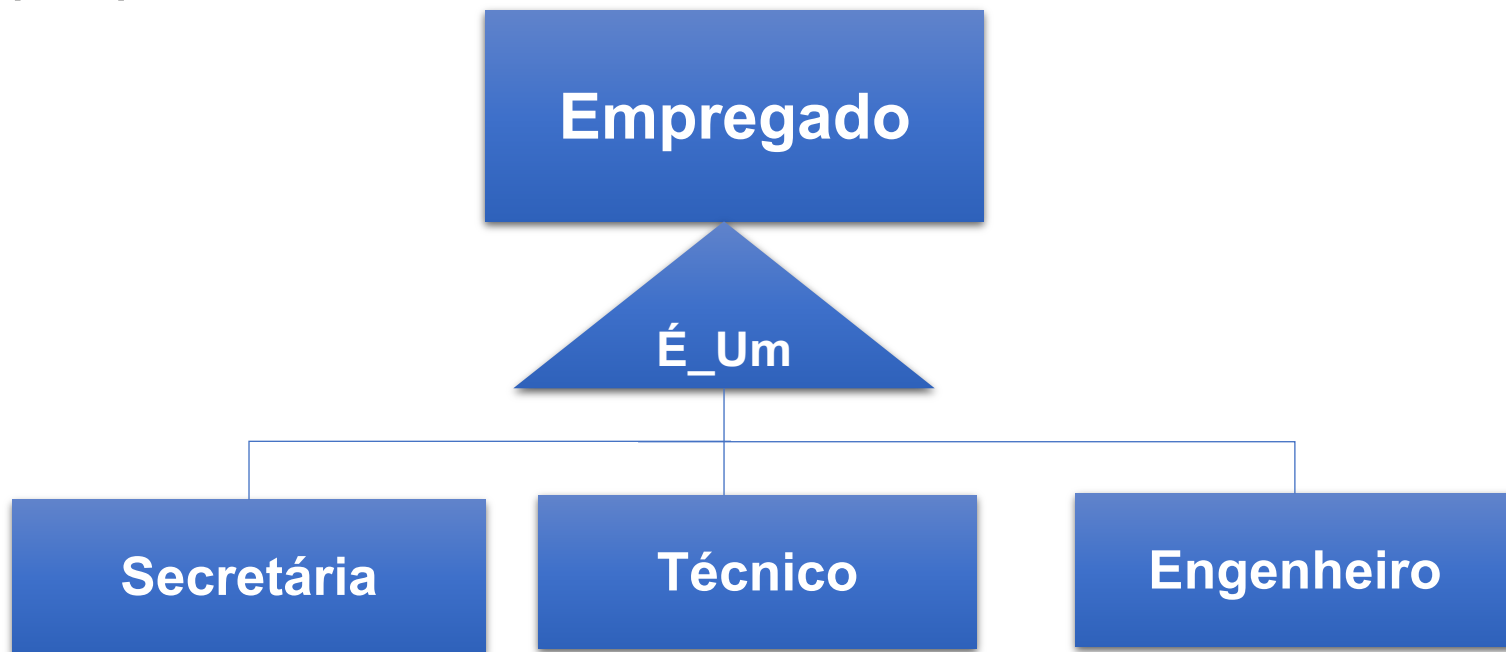
Caminhão(Identificação , Licença, Preço, Num_Eixos, Toneladas)

Regra 7: Generalização e Especialização

C) Criar uma **única relação** L com atributos

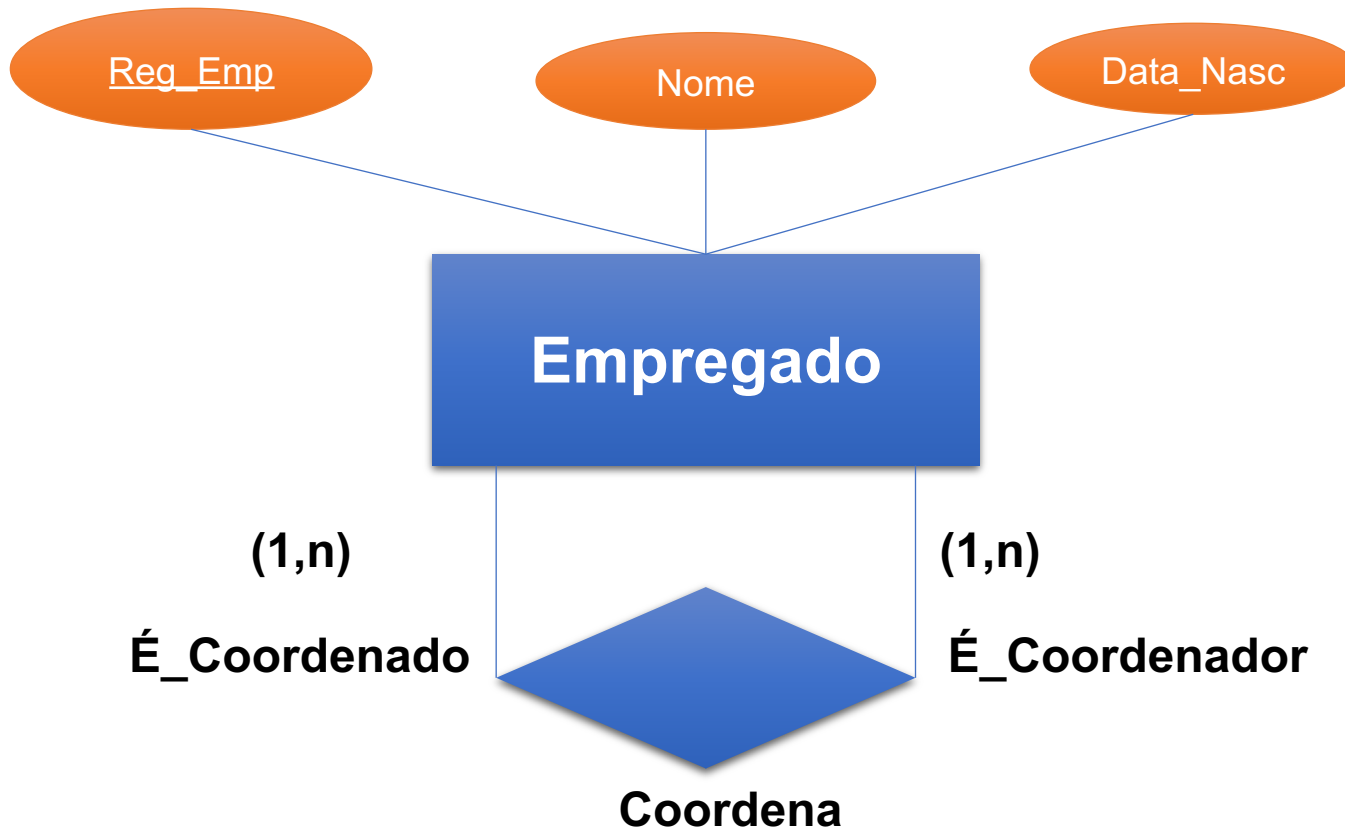
$\text{Atrib}(L) = \{k, a_1, \dots, a_n\} \cup \{\text{atributos de } S_1\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t\}$ e chave k .

Onde t é um atributo de tipo que indica a subclasse a qual a tupla pertence.



Empregado(Matricula, Nome, Salário, Endereço, TipoTrabalho, VelocidadeDigitação, Especialidade, Tipo_Engenheiro, CREA)

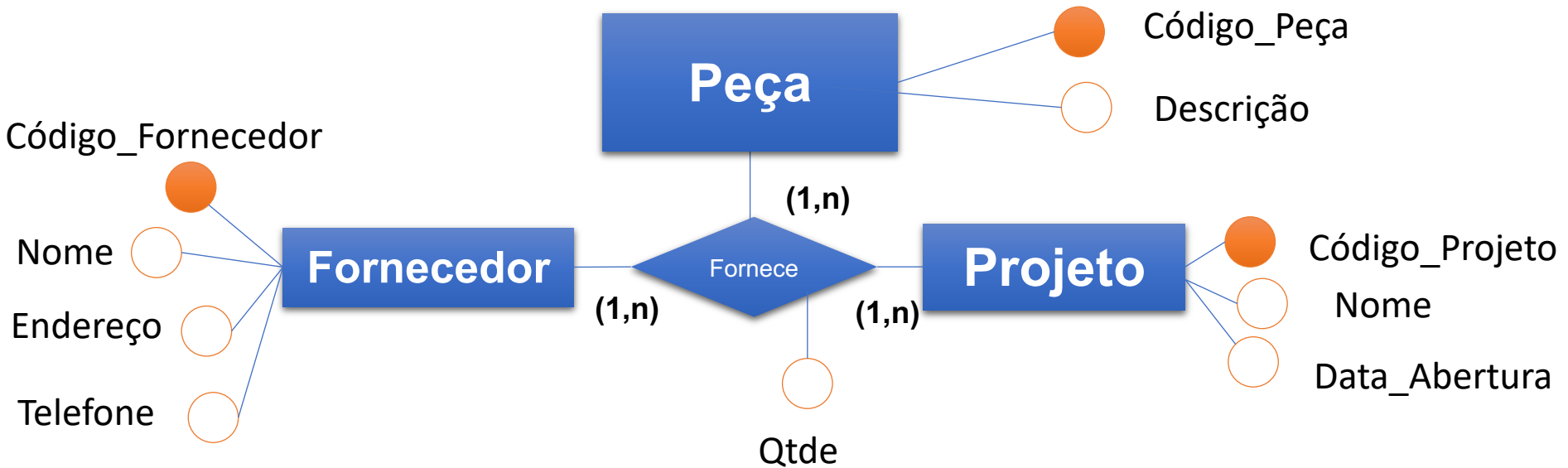
Regra 8: Auto-Relacionamento



Empregado(Reg_Emp, Nome, Data_Nasc)

Coordena(Reg_Subordinado, Reg_Coordenador)

Regra 9: Relacionamentos n-ários



Peça(Código Peça, Descrição)

Fornecedor(Código Fornecedor, Nome, Endereço, Telefone)

Projeto(Código Projeto, Nome, Data_Abertura)

Fornecimento(Código_Peça, Código_Fornecedor, Código_Projeto, Qtde)

Projeto de BD da Disciplina – Parte 2

Passos:

1. Levantamento de Requisitos (Procurar Aplicações)
 2. Modelar aplicações em um SGBD
 - A. Modelo Conceitual (Modelo Entidade-Relacionamento)
 - B. Modelo Lógico (Modelo Relacional – Tabelas)
 3. Implementar no Banco de Dados
 4. Fazer a conexão do Banco de Dados com Java ou C
- (Começar a pensar e fazer o que está de vermelho)

Exemplo de Projeto



Exemplo de Projeto: Modelar uma Livraria (MER) e **Transformar para o Esquema Relacional**

Uma Livraria recebe livros fornecidos por editoras. Os livros são escritos por autor(es). Cada livro é identificado por seu ISBN. Também possui título, preço, ano, e categoria (ex. computação, medicina, etc). Existem vários livros em estoque. O cliente faz um pedido e o acumula na sua cesta de compras. Este pedido pode ser on line ou direto no caixa. O pagamento pode ser: à vista, ou cartão de crédito para pedidos no caixa. E pagamentos on-line só cartão de crédito ou boleto bancário. Em pedidos on-line o envio será por sedex ou encomenda normal.

Exemplo de Projeto

Exemplo de Projeto: Modelar uma Farmácia (MER) e Transformar para o Esquema Relacional

Uma Farmácia possui medicamentos e perfumaria. Os produtos são fornecidos por fabricantes e os fornecedores vendem os produtos, que são enviados por lote. Existem vários tipos de medicamentos. Alguns medicamentos necessitam de receita médica. Há 2 tipos de pagamento: à vista ou no cartão de crédito. A farmácia tem 2 tipos de entrega: no caixa e à domicílio. Faça o MER e expanda usando os requisitos de uma Farmácia.

Transforme o MER em para o esquema Relacional (Tabelas)

