

Disciplina:

# **Banco de Dados Relacional e Não Relacional**

Aula: Projeto de Banco De Dados Relacional

Professor: Anderson Theobaldo



## Nesta Aula

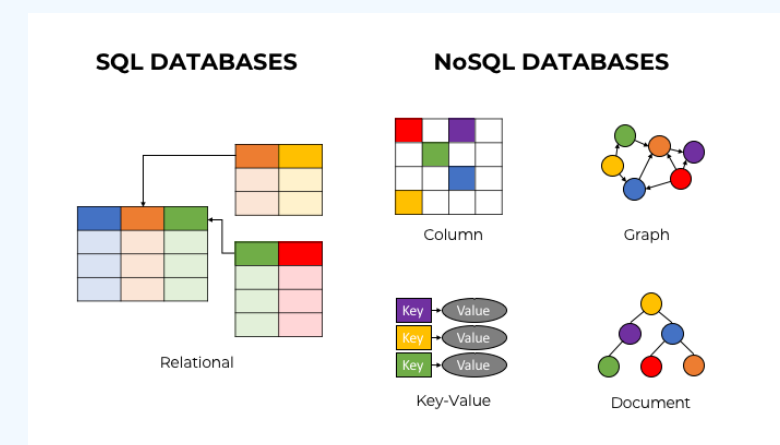
- Modelagem de Dados
- Modelo Conceitual
- Modelo Lógico
- Modelo Físico

# Modelagem de Dados

- É o processo de criação de uma representação abstrata e estruturada dos dados de um sistema de informação.
- Essa representação é projetada para capturar os diferentes tipos de dados que precisam ser armazenados e manipulados pelo sistema.
- A modelagem de dados envolve identificar as entidades relevantes (como pessoas, lugares, coisas, eventos) e os relacionamentos entre elas, bem como os atributos que descrevem essas entidades e relacionamentos.
- Visa facilitar o entendimento sobre os dados e o projeto de dados, ao representarem suas características principais.
- A modelagem de dados é fundamental para o sucesso da implementação de um banco de dados.

# Abstração Dados

Abstração refere-se ao grau de detalhamento dos dados ou conceitos que são descritos no modelo.



## Conceitual:

Descreve apenas parte do banco de dados que são direcionadas para entendimento dos usuários finais.

## Lógico:

Descreve quais dados estão armazenados de fato no banco de dados e as relações que existem entre eles.

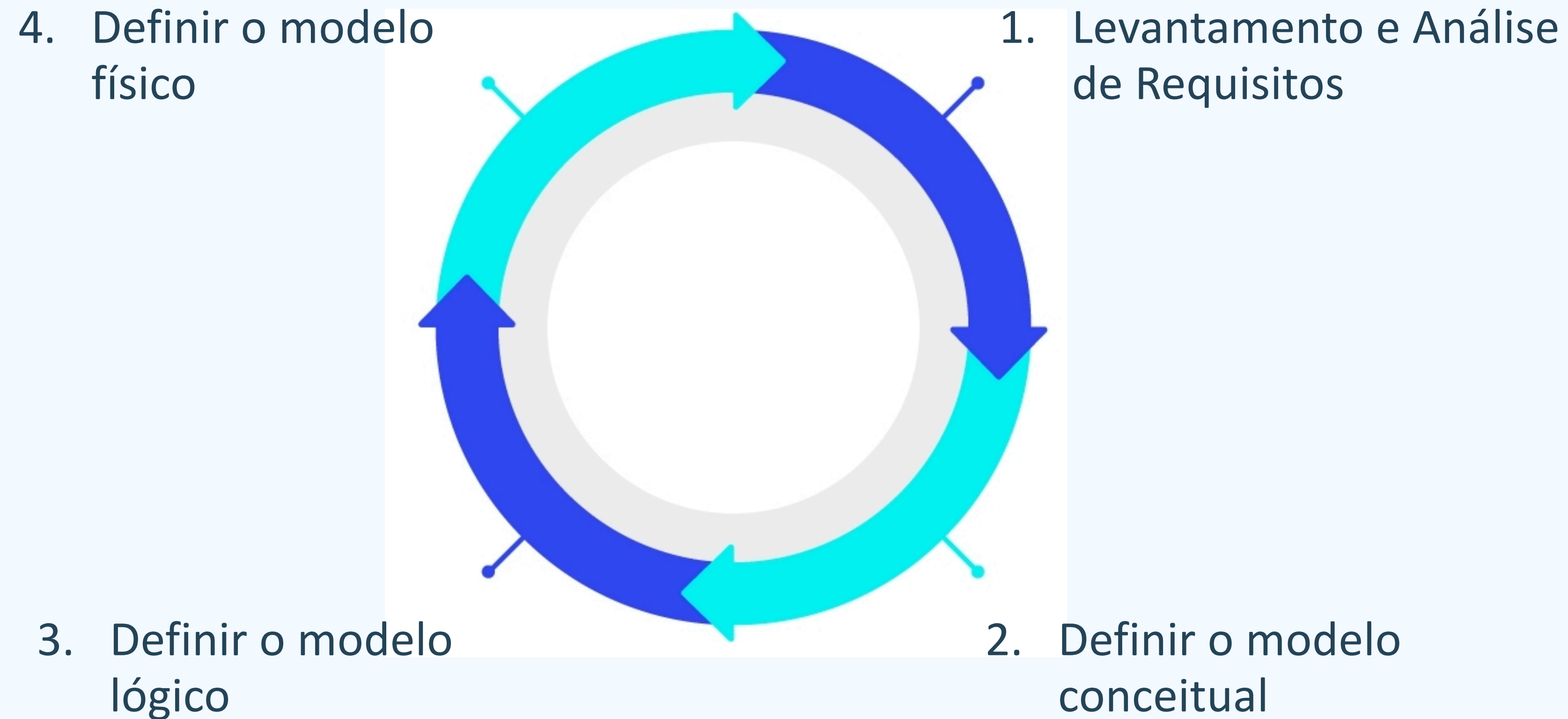
## Físico:

Descreve a estrutura física dos dados e como estão realmente armazenados.

Alto Nível

Baixo Nível

# Projeto de Banco de Dados



# Modelo Conceitual

- Captura os requisitos de informação e regras de negócio sob o ponto de vista do negócio ou do usuário.
- Representa os conceitos no domínio em análise, bem como as associações entre esses conceitos.



# Modelo Entidade-Relacionamento (MER)

- Proposto por Peter Chen em 1976(baseado na teoria relacional de Edgar Frank Codd).
- Baseado na percepção do mundo real.
- Consiste de um conjunto de objetos básicos chamados de entidades, e no relacionamento entre esses objetos.
- Desenvolvido para facilitar o projeto de bancos de dados.
- Este modelo é normalmente apresentado através de um diagrama chamado Diagrama Entidade-Relacionamento (DER).

# Entidade

- É uma representação concreta ou abstrata de um objeto, com características semelhantes, do mundo real.
- Expressões por substantivos ou verbos substantivado.
- Exemplo: produto, cliente, fornecedor, pedido, etc.

## Entidade Forte

Sua existência independe de outra.

Representação gráfica:

Pedido

## Entidade Fraca

Sua existência depende de outra entidade para existir.

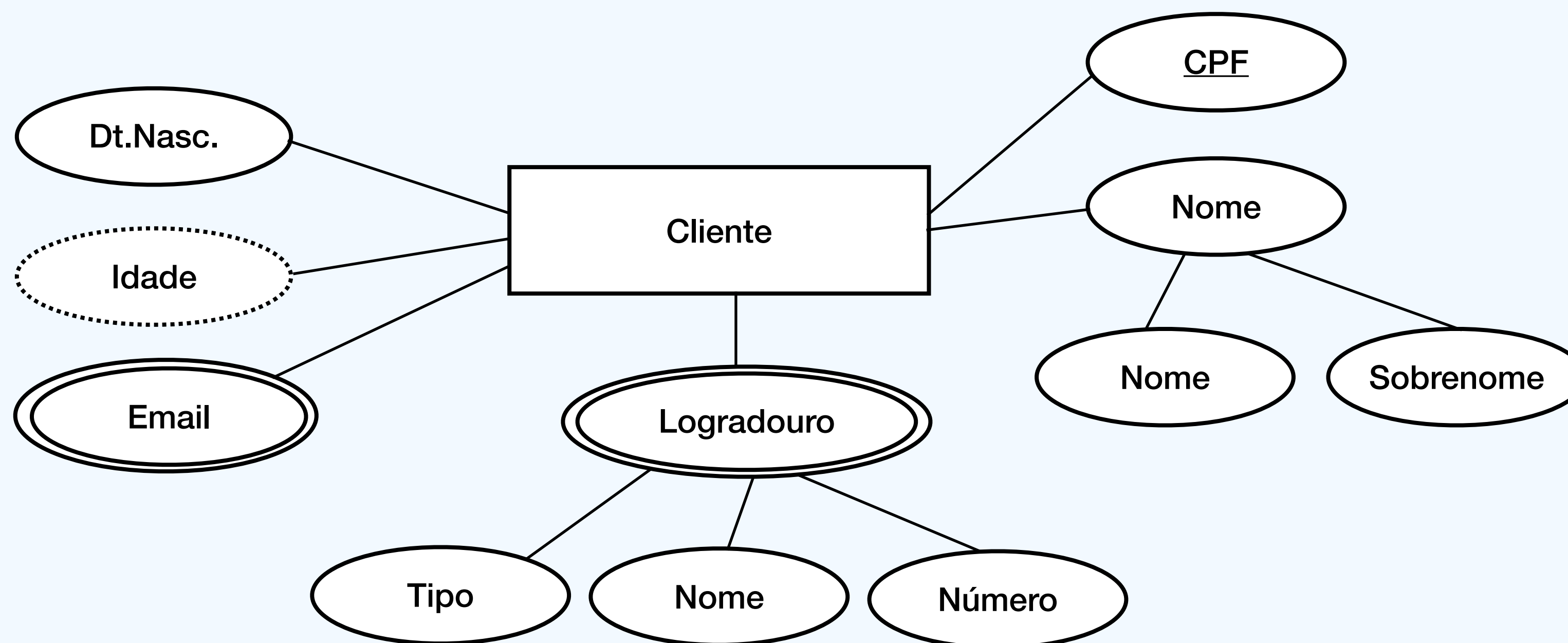
Representação gráfica:

Itens de Pedido



# Atributo

- Propriedades ou características de um tipo de entidade.
  - Exemplo: Nome do cliente, data do pedido, email de contrato, etc.
- Cada atributo terá um valor, que se tornará parte do banco de dados.
- Atributos são representados por ovais contendo seu nome e ligados ao tipo de entidade por linhas retas.



# Classificação dos Atributos



Nome

Atributo simples: são atômicos ou indivisíveis



Email

Atributo multivalorado possui vários valores simultâneos



CPF

Atributo chave identifica de modo único cada instância de uma entidade



Logradouro

Atributo complexo é o agrupamento dos atributos composto e multivalorado.

Tipo

Nome

Número



Nome

Atributo composto é aquele que pode ser dividido em dois ou mais atributos.

P.Nome

Sobrenome



Idade

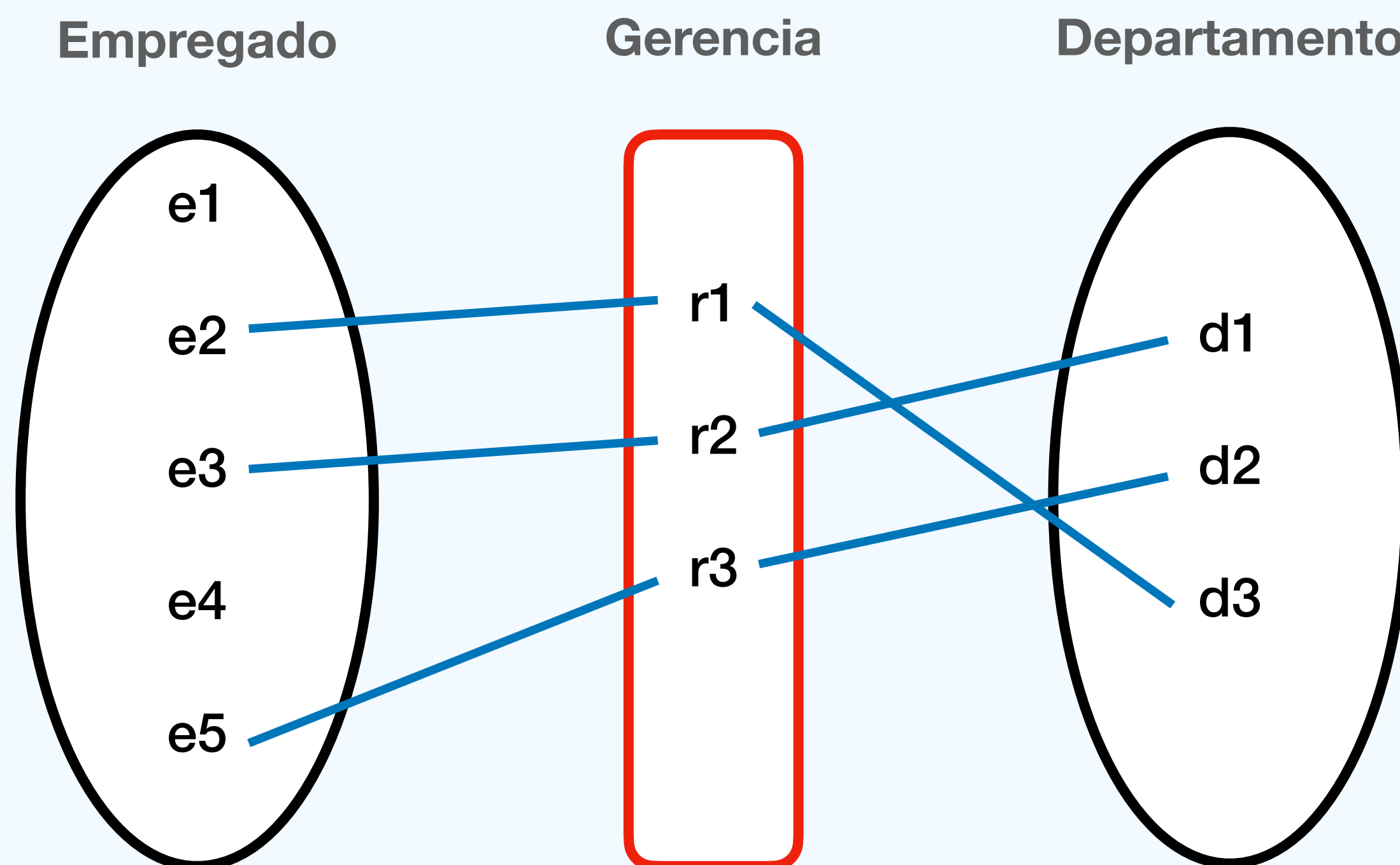
Atributo é derivado quando o valor deste depende do valor de um ou mais atributos.

# Relacionamentos

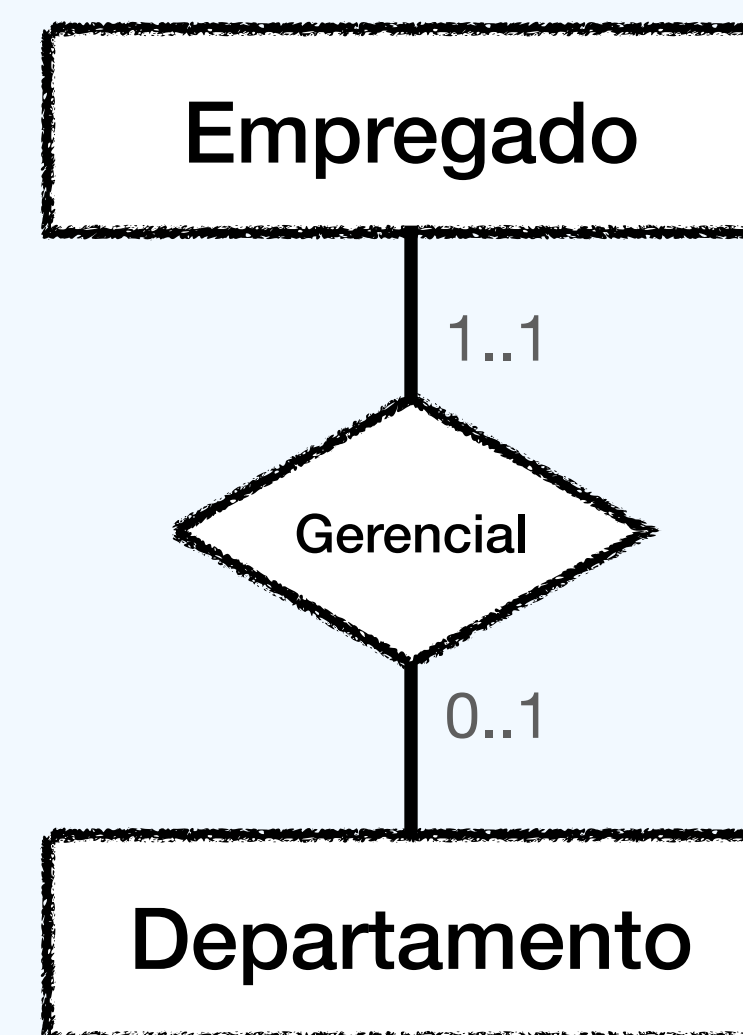
- As entidades são conectadas/relacionadas umas às outra através de relacionamentos.
- Responsáveis por definir as características das ligações entre as entidades.
- Em geral é expresso por um verbo ou por uma locução verbal.
- Restrições dos relacionamentos:
- Cardinalidade: 1:1, 1:N ou N:1, N:N
- Obrigatoriedade: 0 ou 1
- Participação: Total ou Parcial

## Cardinalidade de 1 : 1

Uma instância da entidade A está associada a no máximo uma instância da entidade B, e uma instância da entidade B está associada a no máximo uma instância da entidade A.

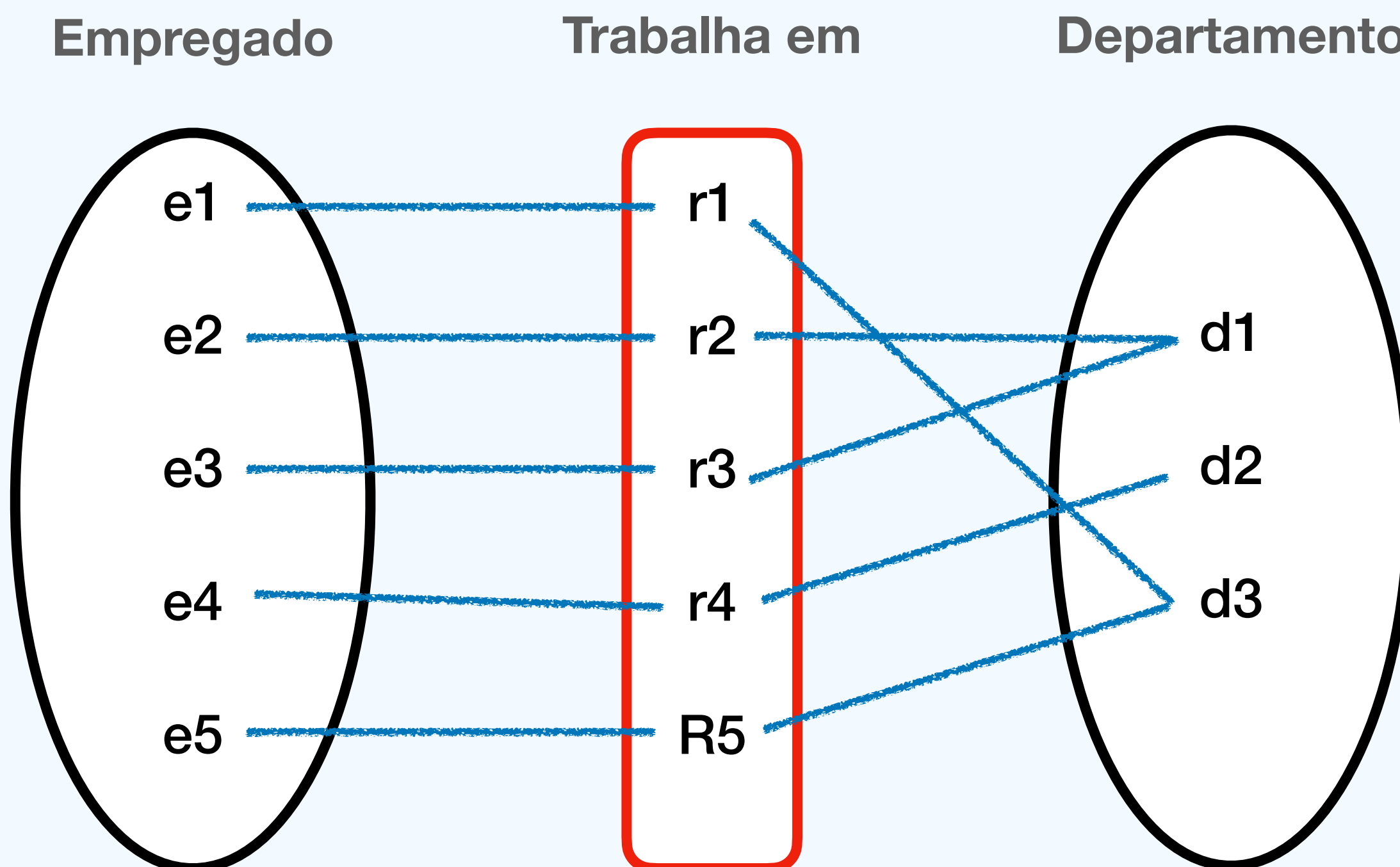


### Representação Gráfica

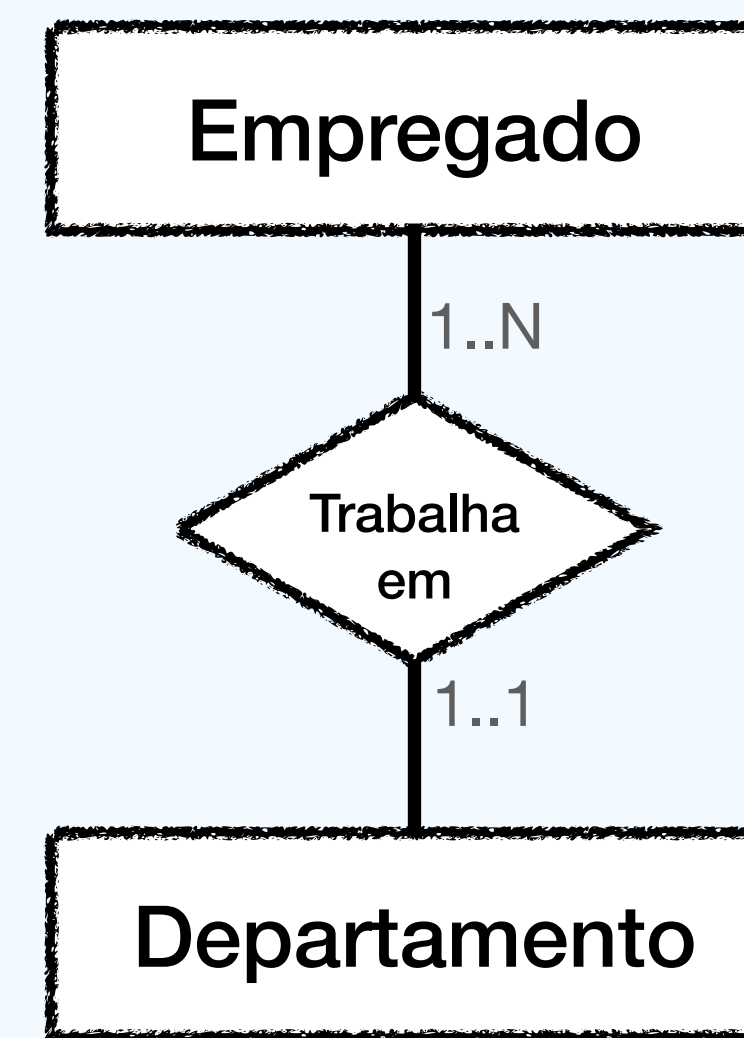


## Cardinalidade de 1 : N ou N : 1

Uma instância da entidade A está associada a várias instâncias da entidade B, e uma instância da entidade B está associada a no máximo uma instância da entidade A.



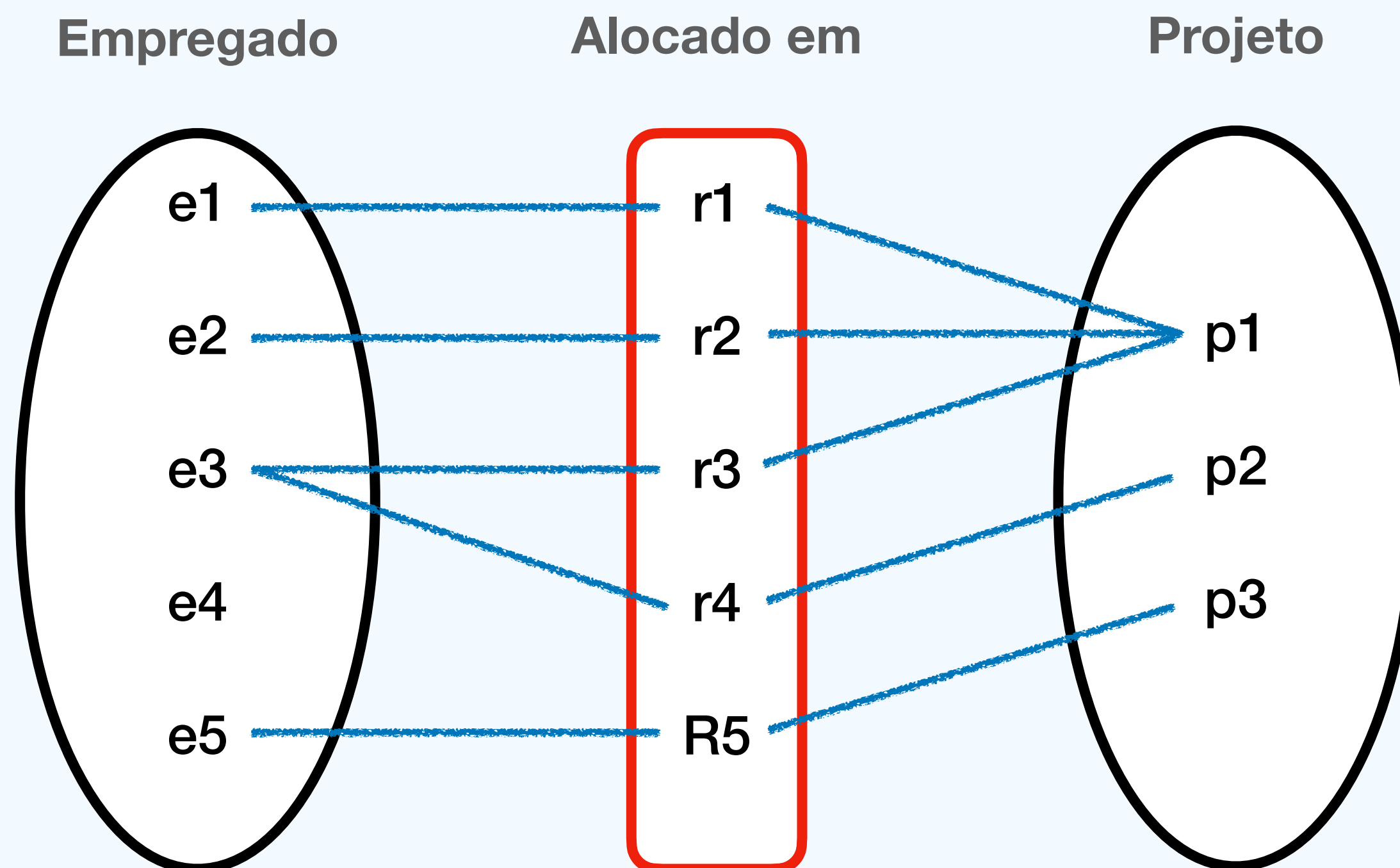
Representação Gráfica



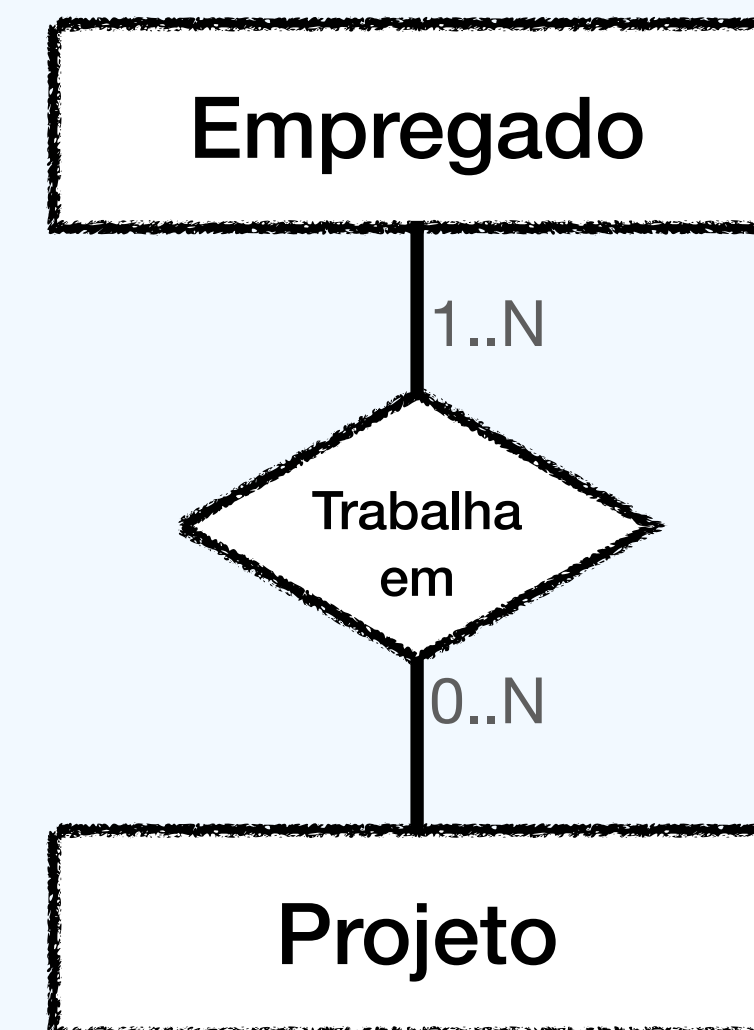


## Cardinalidade de N : N

Muitos para muitos (N:N): Uma instância da entidade A está associada a várias instâncias da entidade B, e uma instância da entidade B está associada a várias instâncias da entidade A.



### Representação Gráfica

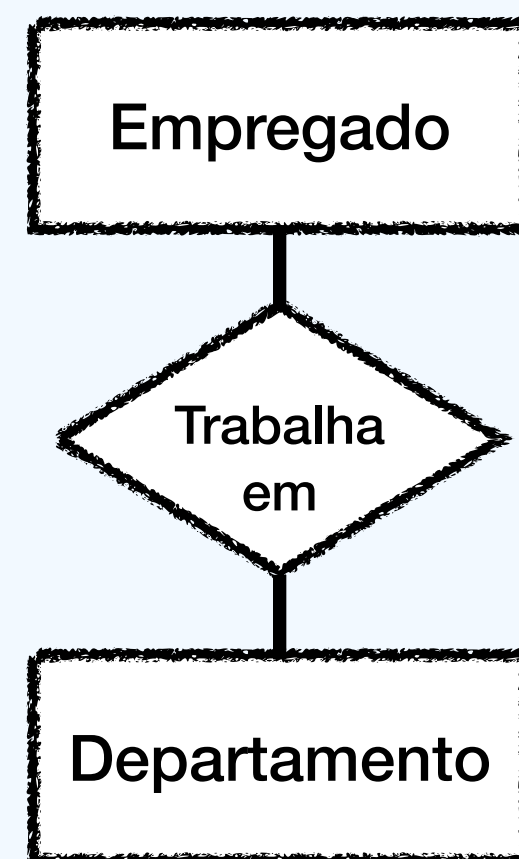




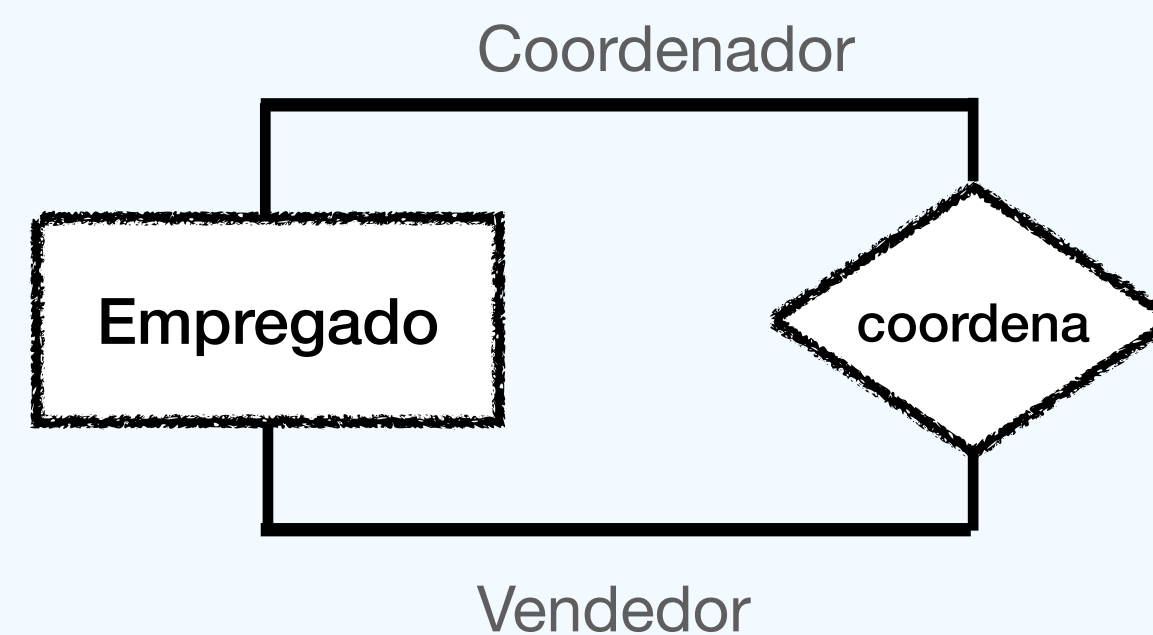
# Grau de Relacionamento

Número de entidades participantes no relacionamento.

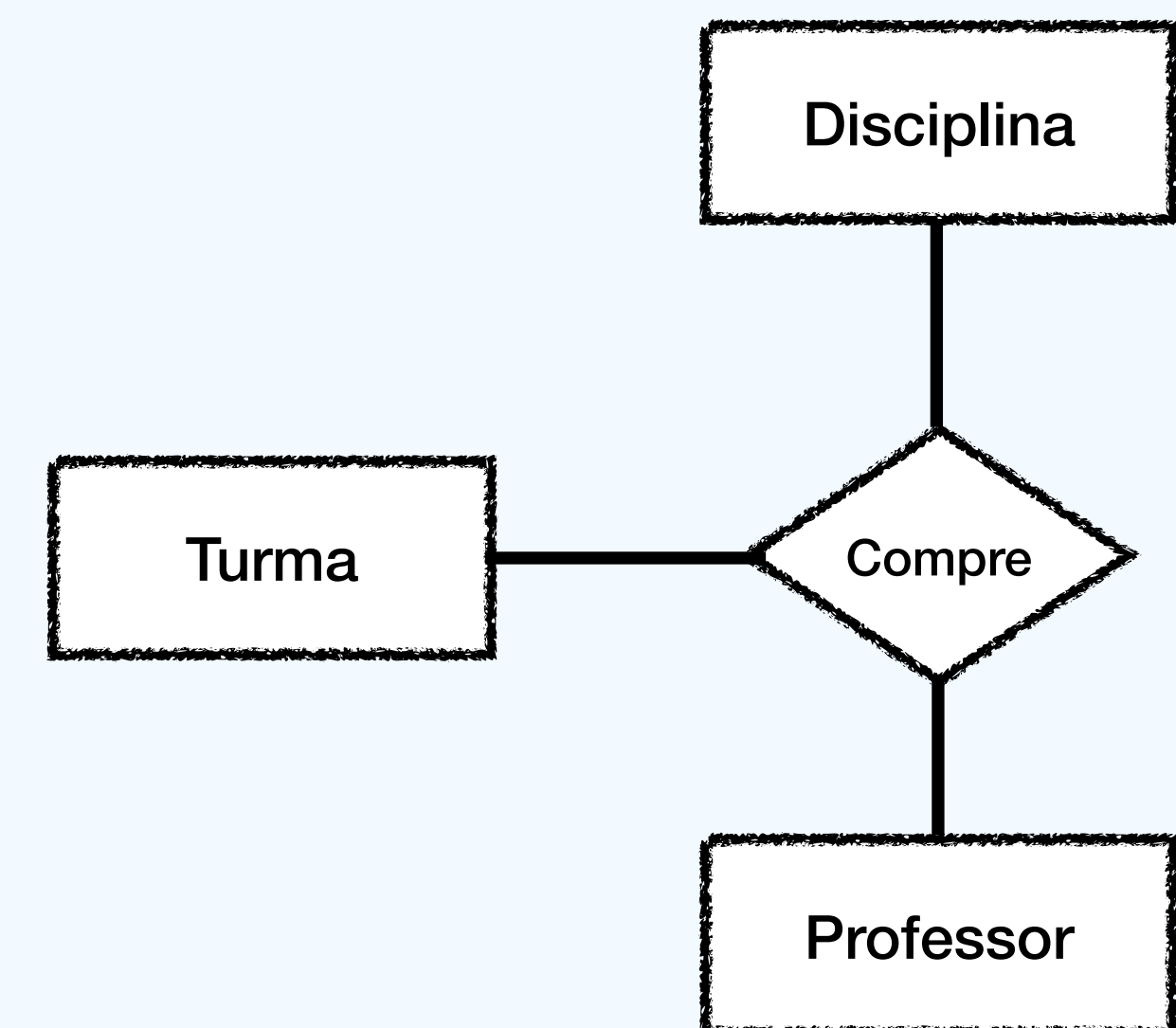
Binários: Envolvem duas entidades.



Binários recursivos: Envolvem a associação entre duas instâncias de uma única entidade as quais participam do relacionamento assumindo diferentes papéis.

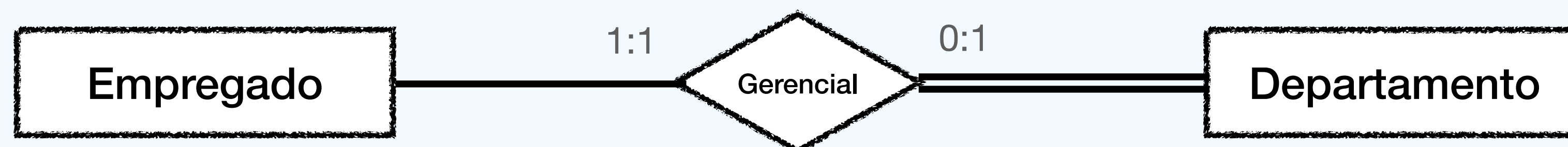


Ternários: Envolvem três entidades.



## Restrição de Participação

- Participação Total: uma relação é total, se toda instância de uma determinada entidade participa do relacionamento na sua totalidade.
  - Exemplo: no relacionamento abaixo, todas as instâncias da entidade Departamento está relacionado a alguma instância da entidade Empregado, porque todo departamento tem que ter um gerente.
- Participação Parcial: uma relação é parcial, se nem toda instância de uma determinada entidade participa do relacionamento.
  - Exemplo: no relacionamento abaixo, nem todas as instâncias da entidade Empregado está relacionado a alguma instância da entidade Departamento, pois somente alguns empregados serão gerentes.



# Construção do Modelo MER

1

Identifique as possíveis entidades e atributos, substantivos e verbos substantivos

2

Determine os tipos de cada atributo(chaves, simples, composto, multivalorados, etc.

3

Identifique e estabeleça os relacionamentos entre as entidades, verbos ou locuções verbais

4

Defina as restrições do relacionamento(cardinalidade, opcionalidade, participação)

## Entidades - Atributos

- Projeto(codigo, nome, local de execução)
  - Departamento(codigo, nome, locais)
  - Empregado(matricula, cpf, nome[nome,sobrenome], logradouro, sexo, data nascimento, salario)
  - Dependente(nome(nome e sobrenome), sexo, data de nascimento e grau de parentesco)
- 
- Relacionamentos
  - Um departamento pode alocar vários funcionários
  - Um departamento é gerenciado por apenas um funcionário(data)
  - Cada departamento controla uma certa quantidade de projetos
  - Os funcionários podem ser alocados em múltiplos projetos (Qte Hs)
  - acompanhar o supervisor direto de cada empregado
  - acompanhar os dependentes diretos de cada empregado

## **Exemplo:**

# **Modelagem para um Sistema de Gerenciamento de Projetos**

- A empresa é organizada em diversos departamentos.
- Cada departamento possui código de identificação, nome e pode estar espalhado em diversos locais.
- Um departamento pode alocar vários funcionários.
- Cada funcionário possui matrícula, nome (nome, sobrenome), CPF, endereço, salário, sexo e data de nascimento registrados.
- Um funcionário pode ser designado para trabalhar em apenas um departamento, durante a vigência de seu contrato.
- Um departamento é gerenciado por apenas um funcionário, e a data em que um funcionário se torna gerente do departamento deve ser registrada.



## **Exemplo:**

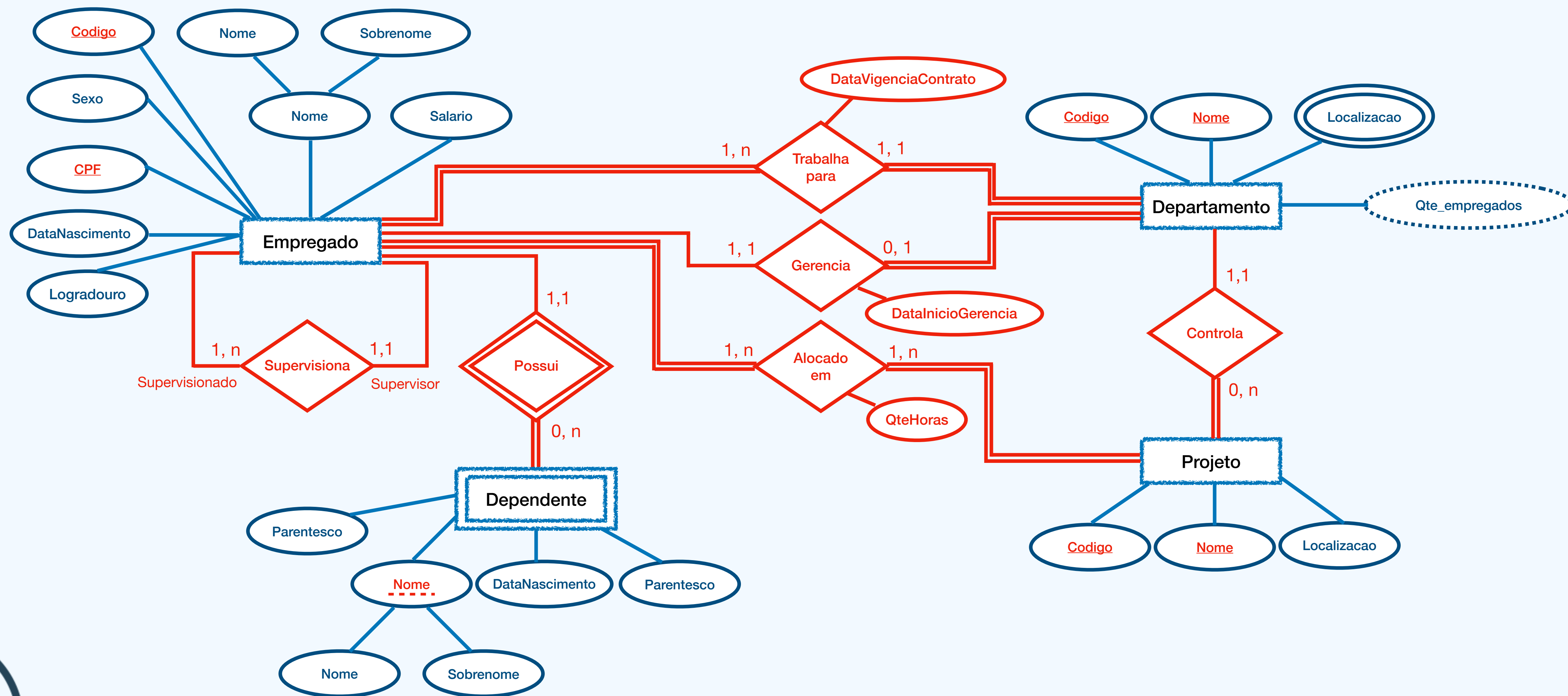
# **Modelagem para um Sistema de Gerenciamento de Projetos**

- Cada departamento controla uma certa quantidade de projetos e cada um deles possuindo código e nome únicos, e local de execução.
- Os funcionários podem ser alocados em múltiplos projetos, com definição de número de horas dedicadas.
- Também é necessário acompanhar o supervisor direto de cada empregado
- É necessário acompanhar os dependentes diretos de cada empregado
- Para cada dependente precisamos saber o nome(nome e sobrenome), sexo, data de nascimento e grau de parentesco com o empregado



# Exemplo:

## Modelagem para um Sistema de Gerenciamento de Projetos



# Modelo Lógico

Modelo de Rede

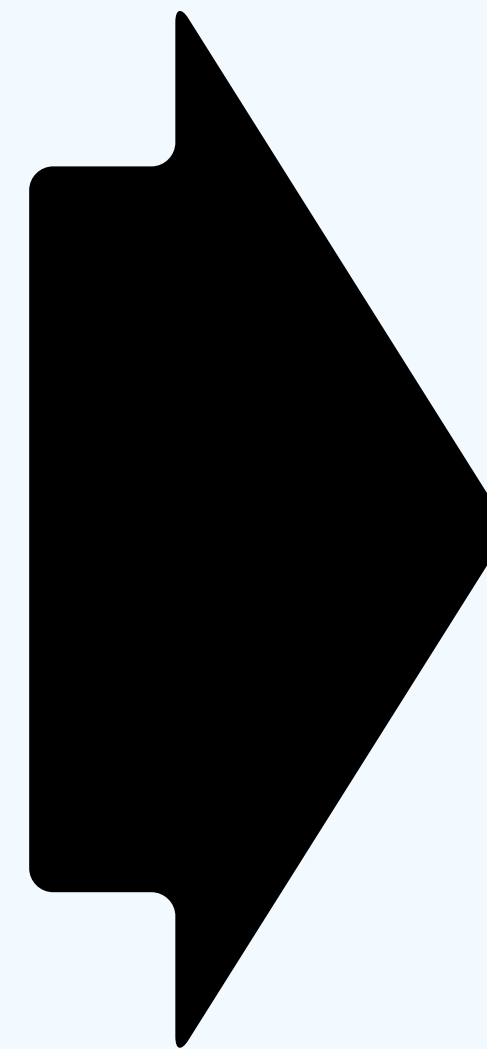
Modelo Hierárquico

Modelo Relacional

Modelo Orientado a Objetos

Modelo a Grafos

Modelo a Documentos



Descreve a estrutura do  
banco de dados conforme  
um paradigma tecnológico

# Modelo Relacional



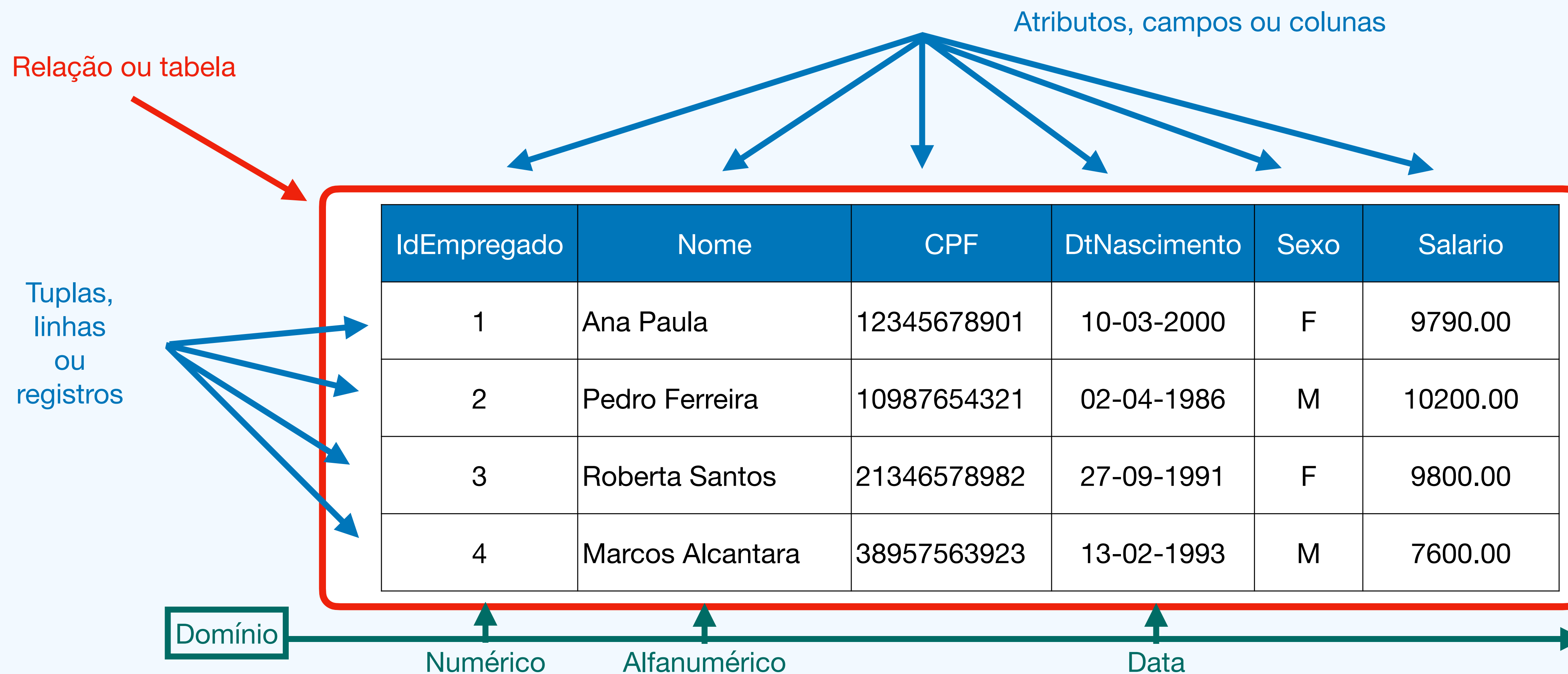
Definido por E.F.Cood em 1970, se tornou o modelo mais usado a partir de meados da década de 1980

Representa um banco de dados com base na teoria de conjuntos e como os conjuntos se relacionam.

Em um banco de dados relacional os dados estão organizados na forma de tabelas, também chamadas de relações.

# Elementos do Modelo Relacional

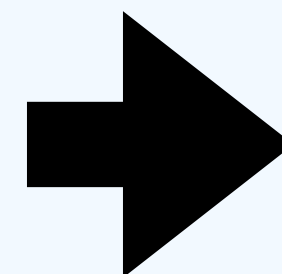
- Tabela(relação) é um conjunto não ordenado de linhas(duplas).
- Cada linha é composta por uma série de campos(colunas ou atributos).
- Cada atributo é definido conforme um domínio específico



# Mapeamento do Modelo Entidade-Relacionamento para o Modelo Lógico(Relacional)

- É possível projetar um esquema relacional a partir de um esquema conceitual
- O resultado materializa o projeto lógico
- O mapeamento é descrito como um algoritmo

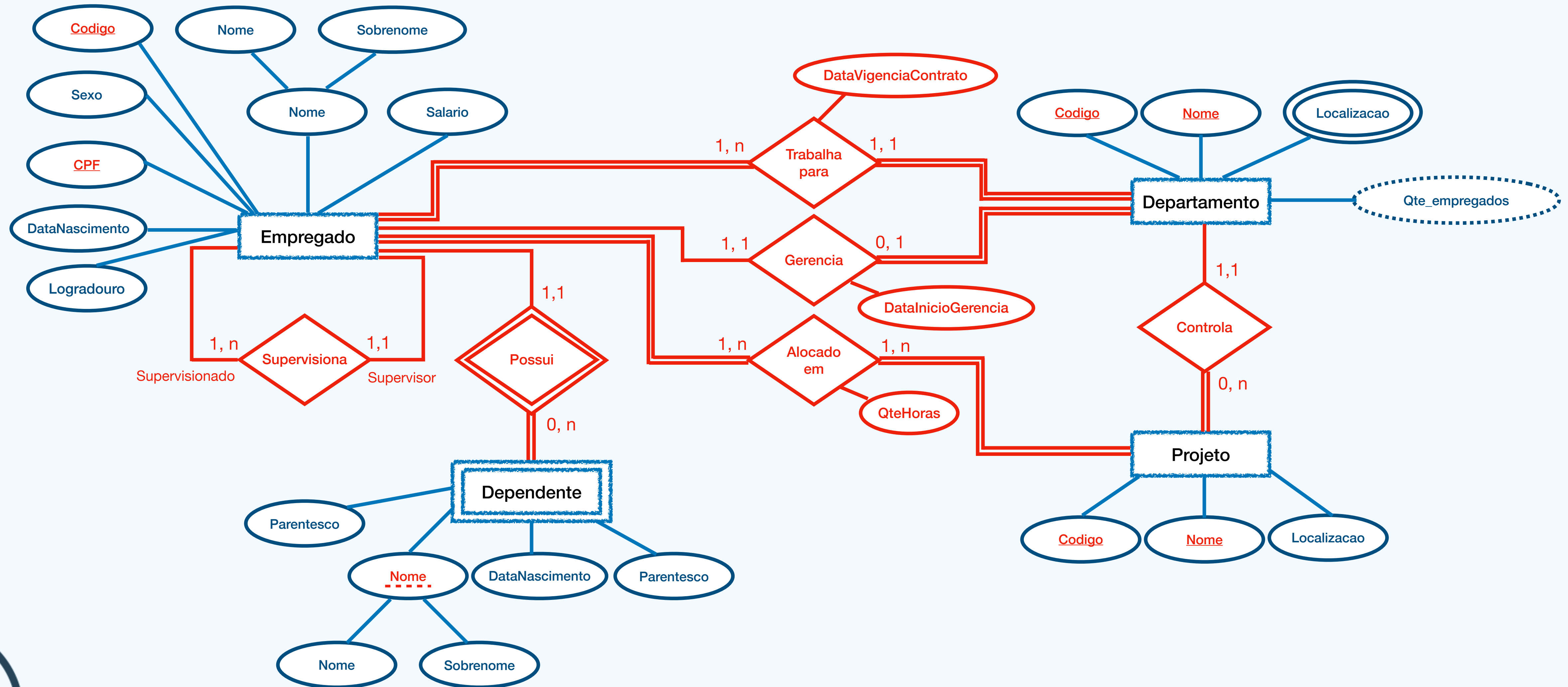
Modelo Conceitual



Modelo Relacional



# Projeto Conceitual - Gerenciamento de Projetos





# Etapas do Algoritmo para o Mapeamento

## Etapa 1: Entidade Forte

- Criar uma relação para cada entidade forte.
- Incluir todos os atributos simples e compostos(incluir todos os atributos simples).
- Escolher o(s) atributo(s) para compor a chave primária.

Empregado
Codigo (PK)
Nome
Sobrenome
CPF
DataNascimento
Sexo
Logradouro
Salario

Projeto
Codigo (PK)
Nome
Localizacao

Departamento
Codigo (PK)
Nome

# Etapas do Algoritmo para o Mapeamento

## Etapa 2: Entidade Fraca

- Criar uma relação para cada entidade fraca.
- Incluir todos os atributos simples e compostos(incluir todos os atributos simples).
- Incluir como chave estrangeira a chave primária da entidade forte relacionada.
- A chave primária será composta pelas chaves estrangeira e parcial.

Dependente
CodigoEmpregado (PK, FK)
Nome (PK)
Sobrenome (PK)
DataNascimento
Parentesco
Sexo



Empregado
Codigo (PK)
Nome
Sobrenome
CPF
DataNascimento
Sexo
Logradouro
Salario

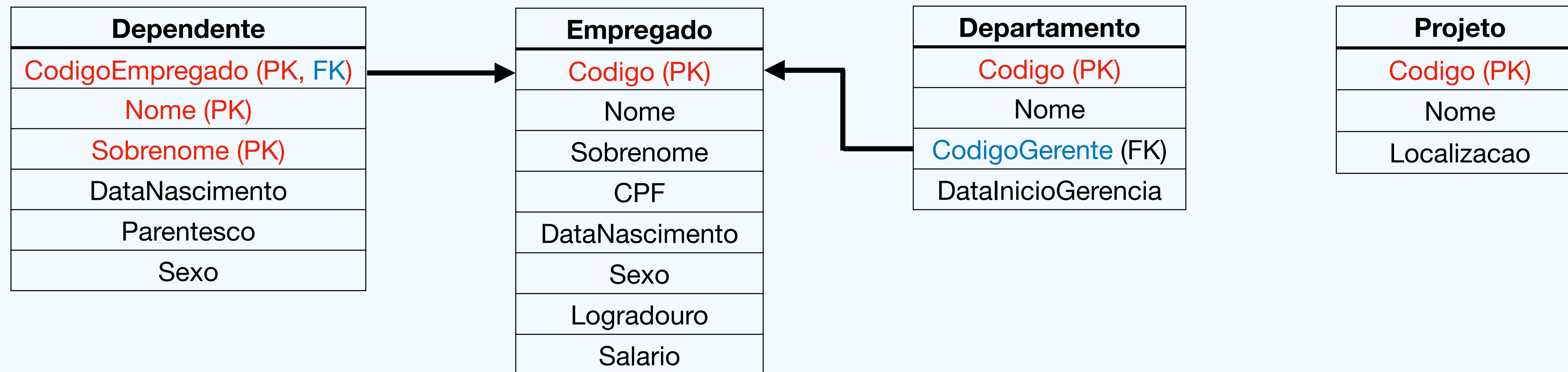
Projeto
Codigo (PK)
Nome
Localizacao

Departamento
Codigo (PK)
Nome

# Etapas do Algoritmo para o Mapeamento

## Etapa 3: Relacionamento binário 1:1

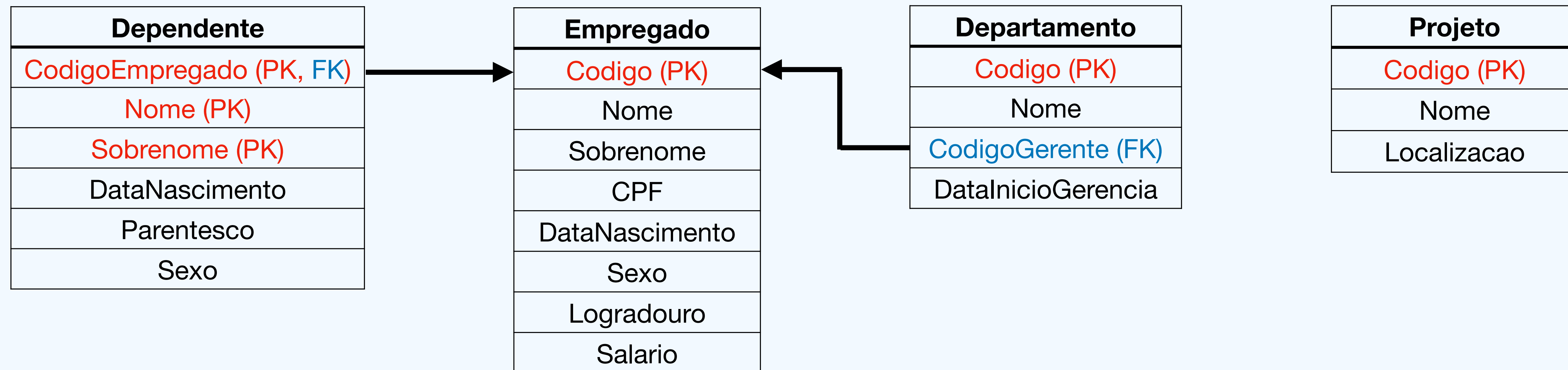
- Incluir como chave estrangeira a chave primária da entidade com participação total no relacionamento.
- Se ambas as entidades tiverem participação total, pode-se unificar as duas entidades em uma única relação.



# Etapas do Algoritmo para o Mapeamento

## Etapa 3: Relacionamento binário 1:1

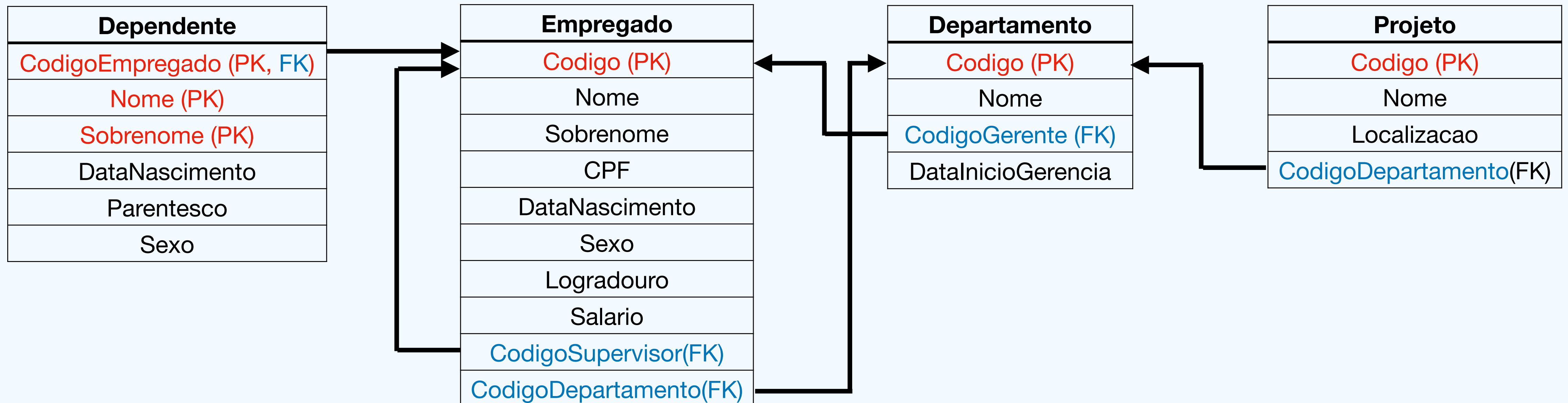
- Incluir como chave estrangeira a chave primária da entidade com participação total no relacionamento.
- Se ambas as entidades tiverem participação total, pode-se unificar as duas entidades em uma única relação.



# Etapas do Algoritmo para o Mapeamento

## Etapa 4:Relacionamento binário 1:N

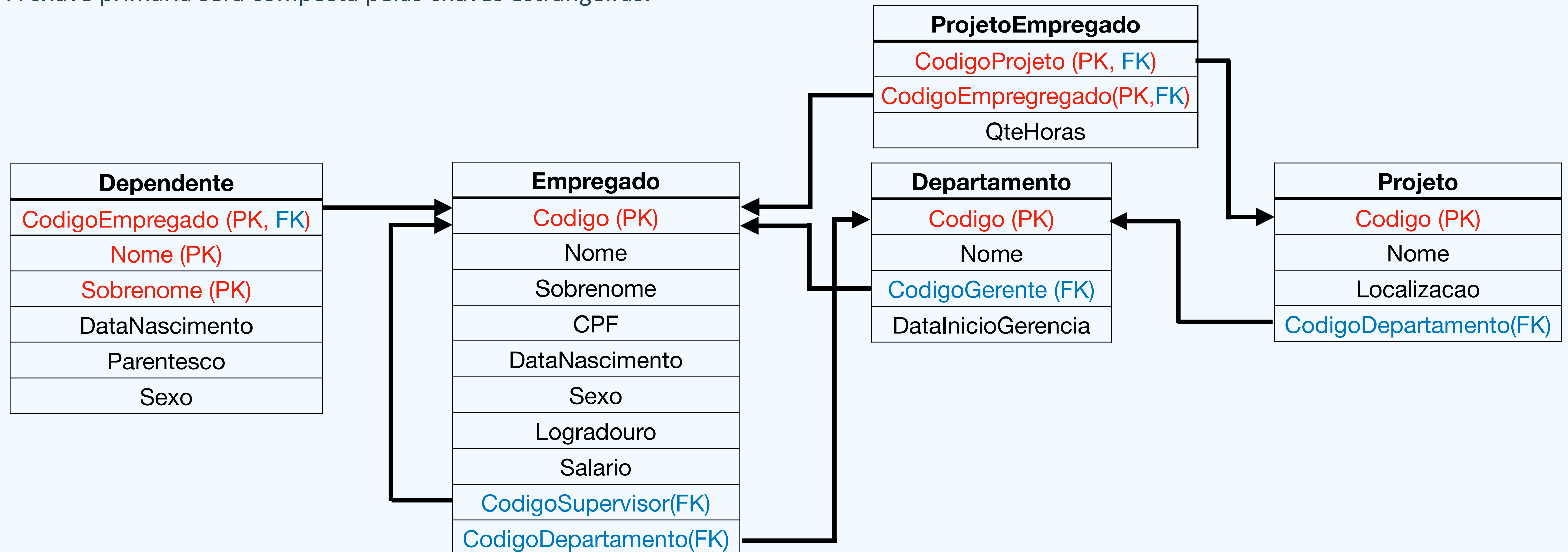
- Na relação que participa do lado N, incluir como chave estrangeira a chave primária da outra relação do relacionamento.



# Etapas do Algoritmo para o Mapeamento

## Etapa 5: Relacionamento Binário N:N

- Para cada relacionamento criar uma nova relação.
- Incluir como chave estrangeira as chaves primárias das relações que representam as entidades participantes.
- Incluir os atributos do relacionamento como atributo da nova relação
- A chave primária será composta pelas chaves estrangeiras.

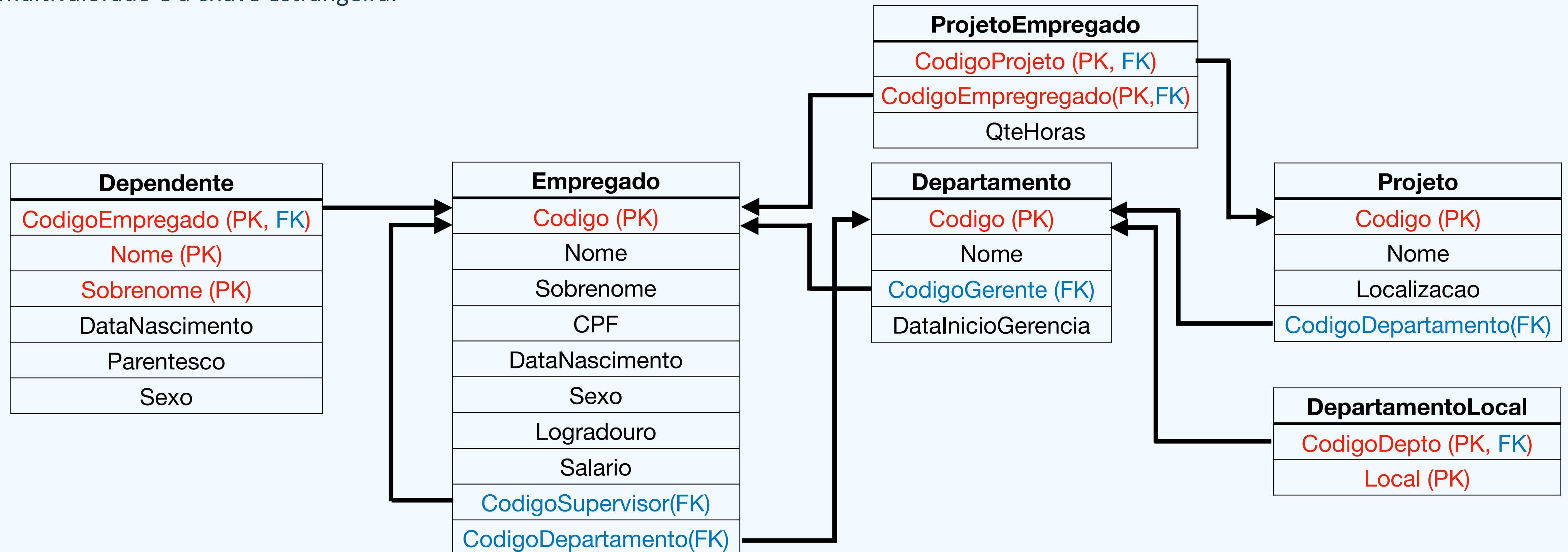




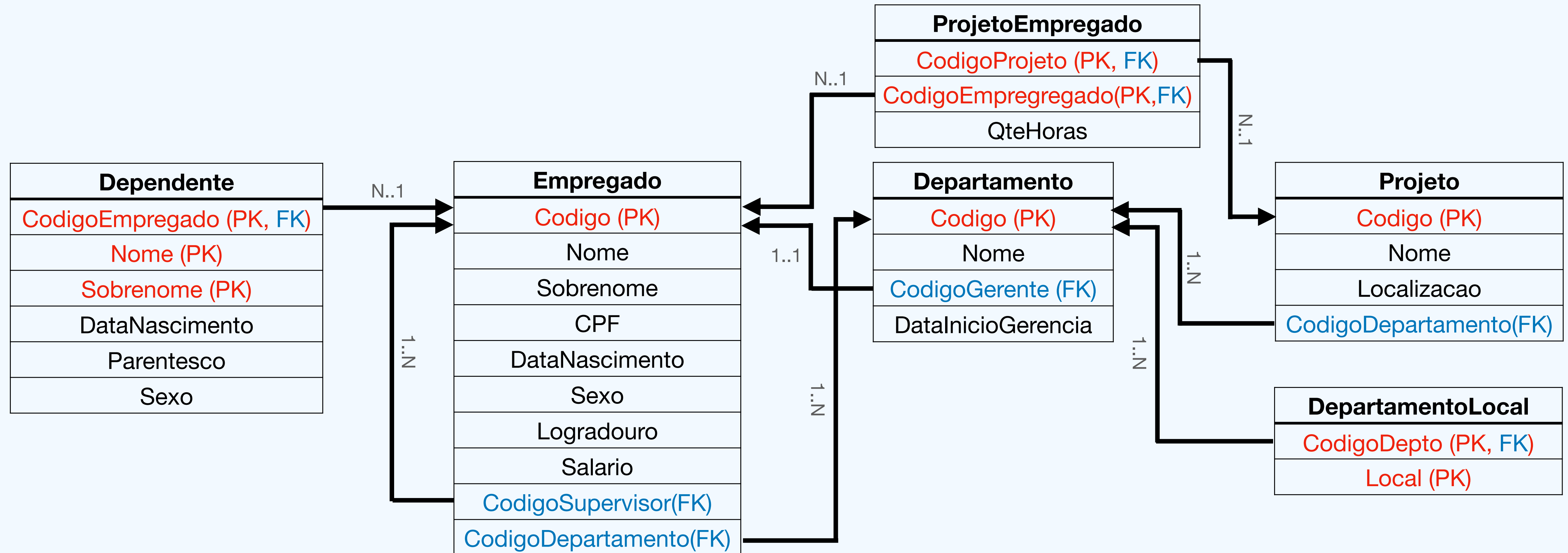
# Etapas do Algoritmo para o Mapeamento

## Etapa 6: Atributo Multivalorado

- Criar uma nova relação incluindo o atributo multivalorado e a chave primária da relação correspondente(será uma chave estrangeira). Se o atributo multivalorado for composto, incluir todos os atributos simples.
- A chave primária será composta pela junção do atributo multivalorado e a chave estrangeira.



# Modelo Relacional Final



# Modelo Físico

- O modelo físico de banco de dados consiste no mapeamento do modelo lógico para um SGBD real.
- Deve levar em conta fatores como:
  - Desempenho
  - Tempo de resposta das transações
  - Alocação de espaço em disco

# Modelo Físico

- O modelo físico de banco de dados consiste no mapeamento do modelo lógico para um SGBD real.
- Deve levar em conta fatores como:
  - Desempenho
  - Tempo de resposta das transações
  - Alocação de espaço em disco

# Modelo Físico

- Principais definições
  - Escolher SGBD
  - Definir estrutura do banco de dados (Schema)
  - Definir índices
  - Definir tamanhos de bloco (páginas)
  - Definir localização física de arquivos e índices
  - Definir permissões de acesso (grupos, usuários e papéis)



# Implementação da Estrutura do Banco de Dados

- A implementação do modelo físico é feita através da linguagem SQL.
- SQL é uma linguagem padronizada para manipulação de bancos de dados.
- Surgiu na década de 70 através da IBM
- Teve algumas variações, decorrentes do trabalho de outros fabricantes de SGBDs.
- Foi padronizada em 1986 (SQL1) pelo ANSI e pela ISO

# Classificação da Linguagem SQL

## Comandos SQL

### DDL

- Create
- Alter
- Drop
- Truncate
- Rename

Usada para definição dos esquemas.

### DML

- Insert
- Update
- Delete

Usada para manipulação dos dados ( inclusão, alteração e exclusão).

### DQL

- Select

Usada para recuperação de dados.

### DCL

- Grant
- Revoke

Usada para conceder e retirar privilégios de usuários em objetos de banco de dados.

### TCL

- Commit
- Rollback
- Save point

Usada para lidar com transações no banco de dados.

# Exemplo de Dicionário de Dados

- Relação: DEP\_Departamento
- Objetivo: Manter o cadastro de departamentos da empresa.

Atributo	Tipo de Dado	Tamanho	Nulo?	Restrições	Observações
DEP_Codigo	Numérico	3	N	PK	Incremental
DEP_Nome	Texto	30	N	UQ	Não pode ter repetição
DEP_EMP_CodigoGerente	Numérico	5	S	FK (EMP_EPREGADO.EMP_Codigo)	
DEP_DataInicioGerencia	Data	8	S		

- Relação: LOC\_Localidade
- Objetivo: Manter o cadastro de localidade de departamentos da empresa.

Atributo	Tipo de Dado	Tamanho	Nulo?	Restrições	Observações
LOC_Id	Numérico	3	N	PK	Incremental
LOC_Nome	Texto	30	N	UQ	Não pode ter repetição

- Relação: DLO\_DepartamentoLocal
- Objetivo: Manter o cadastro de localidade de departamentos da empresa.

Atributo	Tipo de Dado	Tamanho	Nulo?	Restrições	Observações
DLO_DEP_Codigo	Numérico	3	N	PK FK (DEP_Departamento.DEP_Codigo)	
DLO_LOC_Codigo	Numérico	5	N	PK FK (LOC_Localidade.LOC_Codigo)	

# Exemplo de Dicionário de Dados

- Relação: EMP\_Empregado
- Objetivo: Manter o cadastro de empregados envolvidos na execução de algum projeto.

Atributo	Tipo de Dado	Tamanho	Nulo?	Restrições	Observações
EMP_Codigo	Numérico	5	N	PK	Incremental
EMP_CPF	Texto	11	N	UK	
EMP_Nome	Texto	20	N		
EMP_Sobrenome	Texto	30	N		
EMP_Logradouro	Texto	60	S		
EMP_DataNascimento	Data	8	N		
EMP_Salario	Numérico	8,2	N	Não pode ser negativo	
EMP_Sexo	Texto	1	N	Domínio válido = F, M e O	
EMP_DEP_Codigo	Numérico	3	S	FK (DEP_Departamento.DEP_Codigo)	
EMP_CodigoSupervisor	Texto	11	S	FK (EMP_Empregado.EMP_Codigo)	

- Relação: PRJ\_Projeto
- Objetivo: Armazenar os dados dos projetos executados na empresa.

Atributo	Tipo de Dado	Tamanho	Nulo?	Restrições	Observações
PRJ_Codigo	Numérico	5	N	PK	Incremental
PRJ_Nome	Texto	30	N	UQ	
PRJ_Local	Texto	40	N		
PRJ_DEP_Codigo	Numérico	3	N	FK(DEP_DEPARTAMENTO.DEP_Codigo)	

# Exemplo de Dicionário de Dados

- Relação: DPD\_Dependente
- Objetivo: Manter o cadastro de dependentes dos empregados. Será utilizado no cálculo de imposto de renda devido pelo empregado.

Atributo	Tipo de Dado	Tamanho	Nulo?	Restrições	Observações
DPD_Codigo	Numérico	5	N	PK	Incremental
DPD_EMP_Codigo	Numérico	5	N	PK, FK(EMP_EMPREGADO.EMP_Codigo)	
DPD_Nome	Texto	20	N	PK	
DPD_Sobrenome	Texto	30	N	PK	
DPD_Sexo	Texto	1	N	Domínio válido = F, M e O	
DPD_Parentesco	Texto	2	N	Domínio válido = FI, CJ, OU Valor padrão = FI	FI - Filho CJ - Cônjuge OU - Outro

- Relação: PEM\_ProjetoEmpregado
- Objetivo: Cadastrar a relação de horas trabalhadas dos empregados nos respectivos projetos.

Atributo	Tipo de Dado	Tamanho	Nulo?	Restrições	Observações
PEM_PRJ_Codigo	Numérico	5	N	PK FK(PRJ_PROJETO.PRJ_id)	
PEM_EMP_Codigo	Numérico	5	N	PK FK(EMP_EMPREGADO.EMP_id)	
PEM_QteHoras	Numérico	5	N	CK(PEM_QteHs > 0)	Deve ser maior que zero



# Exemplo de Implementação de Estrutura do Banco de Dados utilizando instruções DDL

```
⊖ CREATE TABLE dep_departamento (  
    dep_codigo SERIAL PRIMARY KEY,  
    dep_nome VARCHAR(30) NOT NULL,  
    dep_emp_codigogerente INT,  
    dep_datainiciogerencia DATE,  
    CONSTRAINT dep_uk_nome  
        UNIQUE (dep_nome)  
);  
⊖  
-- Tabela loc_Localidade  
CREATE TABLE loc_Localidade (  
    loc_codigo SERIAL PRIMARY KEY,  
    loc_nome VARCHAR(30) NOT NULL,  
    CONSTRAINT loc_uk_nome  
        UNIQUE (loc_nome)  
);  
⊖  
-- Tabela dlo_departamentoLocal  
CREATE TABLE dlo_departamentoLocal (  
    dlo_dep_codigo INT NOT NULL,  
    dlo_loc_codigo INT NOT NULL,  
    CONSTRAINT dlo_pk_Id  
        PRIMARY KEY (dlo_dep_codigo, dlo_loc_codigo),  
    CONSTRAINT dlo_fk_dep_codigo  
        FOREIGN KEY (dlo_dep_codigo)  
        REFERENCES dep_departamento (dep_codigo),  
    CONSTRAINT dlo_fk_loc_codigo  
        FOREIGN KEY (dlo_loc_codigo)  
        REFERENCES loc_Localidade (loc_codigo)  
);
```

# Exemplo de Implementação de Estrutura do Banco de Dados utilizando instruções DDL

```
-- Tabela emp_empregado
CREATE TABLE emp_empregado (
    emp_codigo SERIAL PRIMARY KEY,
    emp_nome VARCHAR(20) NOT NULL,
    emp_sobrenome VARCHAR(30) NOT NULL,
    emp_cpf CHAR(11),
    emp_logradouro VARCHAR(60) NULL,
    emp_datanascimento DATE NOT NULL,
    emp_salario DECIMAL(8,2) NOT NULL,
    emp_sexo CHAR(1) NOT NULL CHECK (emp_Sexo IN ('F', 'M', 'O')),
    emp_dep_codigo INT NOT NULL,
    emp_codigosupervisor INT,
    CONSTRAINT emp_uk_CPF
        UNIQUE (emp_CPF),
    CONSTRAINT emp_fk_dep_codigo
        FOREIGN KEY (emp_dep_codigo)
        REFERENCES dep_departamento (dep_codigo),
    CONSTRAINT emp_fk_IdSupervisor
        FOREIGN KEY (emp_codigoSupervisor)
        REFERENCES emp_empregado (emp_codigo),
    CONSTRAINT emp_ck_Salario
        CHECK (emp_Salario > 0)
);

-- Tabela dpd_Dependente
CREATE TABLE dpd_Dependente (
    dpd_codigo SERIAL,
    dpd_emp_codigo INT NOT NULL,
    dpd_nome VARCHAR(20) NOT NULL,
    dpd_sobrenome VARCHAR(30) NOT NULL,
    dpd_datanascimento DATE NOT NULL,
    dpd_sexo CHAR(1) NOT NULL CHECK (dpd_Sexo IN ('F', 'M', 'O')),
    dpd_parentesco CHAR(2) NOT NULL
        DEFAULT 'FI'
        CHECK (dpd_Parentesco IN ('FI', 'CJ', 'OU')),
    CONSTRAINT dpd_pk_Id
        PRIMARY KEY (dpd_codigo, dpd_emp_codigo),
    CONSTRAINT dpd_fk_emp_codigo
        FOREIGN KEY (dpd_emp_codigo)
        REFERENCES emp_empregado (emp_codigo)
);
```



# Exemplo de Implementação de Estrutura do Banco de Dados utilizando instruções DDL

```
⊖ -- Tabela prj_Projeto
CREATE TABLE prj_Projeto (
    prj_codigo SERIAL PRIMARY KEY,
    prj_nome VARCHAR(60) NOT NULL,
    prj_loc_codigo INT NOT NULL,
    prj_dep_codigo INT NOT NULL,
    CONSTRAINT prj_uk_nome UNIQUE (prj_nome),
    CONSTRAINT prj_fk_dep_codigo
        FOREIGN KEY (prj_dep_codigo)
        REFERENCES dep_departamento (dep_codigo),
    CONSTRAINT prj_fk_loc_codigo
        FOREIGN KEY (prj_loc_codigo)
        REFERENCES loc_localidade (loc_codigo)
);

⊖ -- Tabela pem_Projetoempregado
CREATE TABLE pem_Projetoempregado (
    pem_prj_codigo INT NOT NULL,
    pem_emp_codigo INT NOT NULL,
    pem_qtehoras INT NOT NULL CHECK (pem_qtehoras > 0),
    CONSTRAINT pem_pk_Id PRIMARY KEY (pem_prj_codigo, pem_emp_codigo),
    CONSTRAINT pem_fk_prj_codigo
        FOREIGN KEY (pem_prj_codigo)
        REFERENCES prj_Projeto (prj_codigo),
    CONSTRAINT pem_fk_emp_codigo
        FOREIGN KEY (pem_emp_codigo)
        REFERENCES emp_empregado (emp_codigo)
);

⊖ Alter Table dep_departamento
Add Constraint dep_fk_emp_codigo
Foreign Key(dep_emp_codigogerente)
References emp_empregado(emp_codigo);
```

# Referência Bibliográfica

Elmasri & Navathe. Sistemas de Banco de Dados. Editora Person.

