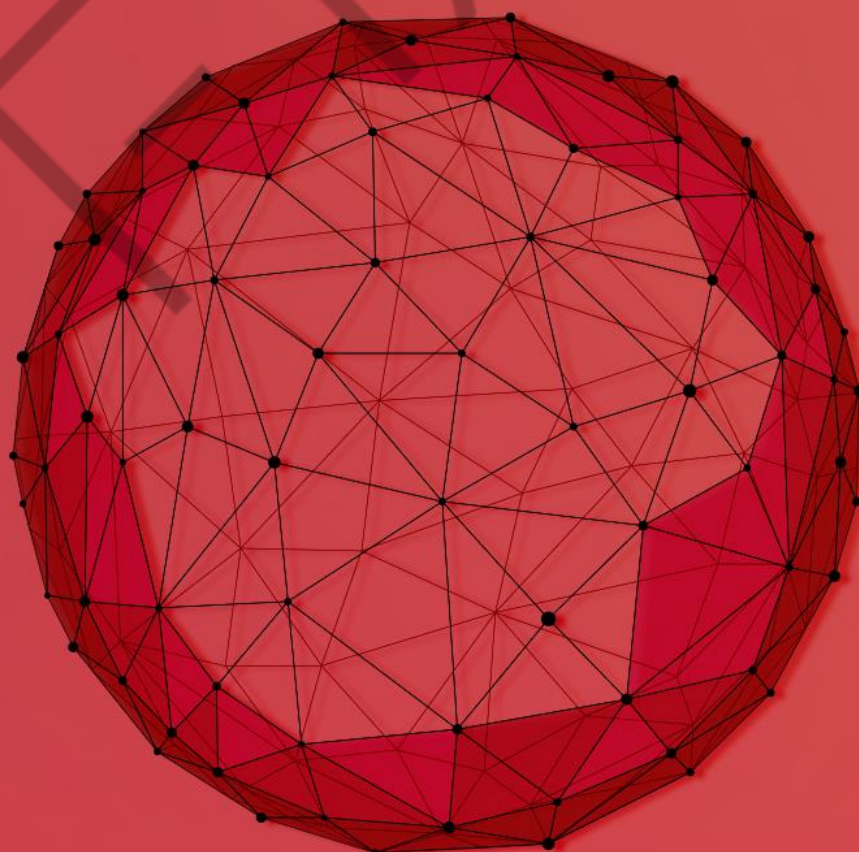


MODELING

# AGREGANDO AO MODELO DE DADOS

RITA DE CÁSSIA RODRIGUES E HENRIQUE POYATOS



05

## LISTA DE FIGURAS

Figura 1 – Exemplo de relacionamento entre médico e paciente .....	7
Figura 2 – Exemplo de relacionamento entre médico e paciente (2) .....	7
Figura 3 – Criação da entidade associativa consulta .....	8
Figura 4 – Característica da entidade Consulta .....	9
Figura 5 – Relacionamento entre médico e paciente em um modelo lógico .....	9
Figura 6 – Relacionamento médico e paciente envolvendo a entidade associativa consulta .....	9
Figura 7 – Exemplo de ocorrências no relacionamento médico, consulta e paciente .....	11
Figura 8 – Relacionamento entre consulta e medicamento .....	12
Figura 9 – Relacionamento entre consulta e medicamento (2) .....	13
Figura 10 – Criação da entidade associativa prescrição entre consulta e medicamento .....	14
Figura 11 – Característica da entidade PRESCRICAO .....	14
Figura 12 – Relacionamento entre consulta e medicamento no Modelo Lógico .....	15
Figura 13 – Relacionamento entre consulta, prescrição e medicamento no Modelo Lógico .....	15
Figura 14 – Classe associativa prescrição com uma chave composta .....	16
Figura 15 – Exemplos de ocorrências das futuras tabelas consulta, prescrição e medicamento .....	16
Figura 16 – Modelo conceitual do relacionamento entre engenheiro e projeto .....	18
Figura 17 – Modelo lógico do relacionamento entre engenheiro e projeto .....	18
Figura 18 – Criação da entidade associativa atuação, entre as entidades engenheiro e projeto .....	19
Figura 19 – Abordagem criando uma chave para a entidade associativa atuação ...	19
Figura 20 – Modelo lógico das entidades professor e disciplina .....	21
Figura 21 – Criação da entidade associativa professor_disciplina .....	21
Figura 22 – Exemplos de ocorrências para pessoas físicas e jurídicas .....	23
Figura 23 – Relacionamento entre pessoa, pessoa física e pessoa jurídica .....	24
Figura 24 – Modelo lógico utilizando um arco, notação de Barker .....	25
Figura 25 – Modelo lógico utilizando um arco, notação de Engenharia da Informação .....	26
Figura 26 – Exemplos de ocorrências entre funcionário, engenheiro, secretária e médico .....	27
Figura 27 – Modelo lógico envolvendo funcionário, engenheiro, secretária e médico, notação de Barker .....	29
Figura 28 – Modelo lógico envolvendo funcionário, engenheiro, secretária e médico, notação de Engenharia da Informação .....	30
Figura 29 – Modelo lógico da entidade funcionário .....	31
Figura 30 – Modelo lógico das entidades médico, engenheiro e secretária .....	32
Figura 31 – Exemplos de ocorrências para as futuras tabelas pessoa, pessoa física e pessoa jurídica .....	33
Figura 32 – Exemplos de ocorrências para as futuras tabelas pessoa, pessoa física e pessoa jurídica (2) .....	34
Figura 33 – Exemplos de ocorrências para as futuras tabelas funcionário, secretária, médico e engenheiro .....	35

Figura 34 – Exemplos de ocorrências para as futuras tabelas funcionário, secretária, médico e engenheiro (2) .....	36
Figura 35 – Exemplos de ocorrências para as futuras tabelas pessoa, professor, funcionário e aluno. ....	37
Figura 36 – Modelo lógico entre pedido, pessoa física e pessoa jurídica .....	38
Figura 37 – Modelo lógico entre pedido, cliente, pessoa física e pessoa jurídica .....	39
Figura 38 – Modelo lógico formalizando algumas entidades.....	40
Figura 39 – Modelo lógico com as entidades coordenador e funcionário.....	43
Figura 40 – Exemplos de ocorrências das futuras tabelas coordenador e funcionário .....	43
Figura 41 – Modelo lógico representando o autorrelacionamento da entidade funcionário.....	44
Figura 42 – Exemplos de ocorrências da futura tabela funcionário com autorrelacionamento.....	45
Figura 43 – Modelo lógico representando o autorrelacionamento da entidade funcionário (2) .....	46
Figura 44 – Modelo lógico das entidades funcionário e relacionamento_funcionario, notação de Barker .....	47
Figura 45 – Modelo lógico das entidades funcionário e relacionamento_funcionario, notação da Engenharia da Informação .....	47
Figura 46 – Exemplos de relacionamentos entre funcionário e relacionamento_funcionario .....	48

## SUMÁRIO

1 AGREGANDO AO MODELO DE DADOS.....	5
1.1 Sobre agregação.....	5
1.1.1 Exemplificando agregação .....	6
1.1.2 Regras ou soluções possíveis para determinar a chave primária em uma agregação .....	10
1.1.3 Evoluindo o exemplo de agregação .....	12
1.1.4 Um pouco mais de atributos do relacionamento .....	17
1.1.5 Exemplificando outra situação na qual temos entidade associativa.....	20
1.2 Sobre herança.....	22
1.2.1 Exemplificando especialização, generalização e herança.....	23
1.2.2 Regras para determinar a chave primária nas entidades especializadas.....	32
1.2.3 Classificação ou Restrições em uma especialização/generalização .....	34
1.2.4 Exemplo de situações em que utilizamos ou não especialização/ generalização .....	37
1.3 Sobre Autorrelacionamento.....	40
1.3.1 Exemplificando relacionamentos recursivos ou autorrelacionamento .....	42
REFERÊNCIAS.....	49

## 1 AGREGANDO AO MODELO DE DADOS

Relacionamentos não são fáceis: pegue como referencial os relacionamentos humanos que são tão cheios de variáveis e contextos. Ao trabalhar na modelagem de dados e começarmos a identificar as entidades e seus relacionamentos, reparamos que estes são muito influenciados pelo contexto de negócio e, ao reproduzir isso, podem se tornar completos.

Este capítulo aprofunda o que foi iniciado no capítulo anterior – abordamos esses tipos de relacionamentos menos triviais, mas extremamente importantes e presentes em nossos sistemas computacionais.

### 1.1 Sobre agregação

Podemos dizer que um modelo de dados pode e deve passar por um processo de refinamento. Observamos algumas situações particulares, que chamamos de extensões, cuja função é possibilitar a correção dessas particularidades para que o modelo de dados possa ser posteriormente implementado de forma física, por meio de um SGBD.

A agregação é uma forma de ajustar a implementação de um relacionamento do tipo muitos para muitos.

Não é possível implementar fisicamente um relacionamento muitos para muitos considerando que uma chave estrangeira não pode ser multivalorada. Trata-se de uma limitação de um sistema de banco de dados relacional: uma chave estrangeira deve fazer referência a um único valor (é monovalorada), conforme vimos no capítulo anterior.

Quando ajustamos um relacionamento do tipo muitos para muitos, observamos que, em boa parte das situações, conseguimos encontrar atributos que caracterizam essa relação (**atributos do relacionamento**).

Como encontramos atributos que caracterizam o relacionamento e ainda há a necessidade de manter a chave estrangeira monovalorada, transformamos o relacionamento e os atributos que caracterizam esse relacionamento em um objeto

agregado. Em outras palavras, criamos uma nova entidade para representar a associação – criamos uma entidade ASSOCIATIVA.

Uma entidade associativa não existe por si só em um modelo entidade relacionamento, sua existência está condicionada à existência de duas ou mais entidades existentes no modelo. Ela existe porque as duas outras entidades existem.

É chamado de agregação, pois a nova entidade agrega atributos das duas entidades que anteriormente se associavam, mais os atributos que caracterizam o relacionamento.

Normalmente encontramos um nome que caracteriza a nova entidade (associativa) em função das necessidades de negócio.

Há situações em que não encontramos atributos que caracterizam o relacionamento em um relacionamento muitos para muitos, nesse caso, utilizamos uma nova entidade com intuito de ligação e solução para as chaves estrangeiras, de modo que sejam monovaloradas. Assim, teremos uma entidade que receberá apenas as chaves estrangeiras oriundas das entidades associadas.

Quando temos entidades apenas de ligação (associativas), normalmente não conseguimos identificar um nome que represente a entidade e, no momento de nomeá-la, utilizamos os nomes das entidades que se associavam anteriormente.

Trabalharemos alguns exemplos ainda neste capítulo, para deixar mais claros esses conceitos.

### **1.1.1 Exemplificando agregação**

Suponha a seguinte situação: que um médico possa avaliar, examinar e consultar um paciente; e que um paciente possa ser avaliado, examinado, consultado por um médico.

Cada médico pode avaliar, examinar, consultar vários pacientes, inclusive o mesmo paciente várias vezes. Cada paciente pode ser avaliado, examinado, consultado por vários médicos, inclusive com o mesmo médico várias vezes.

Podemos representar a associação entre médico e paciente (Figura “Exemplo de relacionamento entre médico e paciente”).



Figura 1 – Exemplo de relacionamento entre médico e paciente  
Fonte: Elaborado pelo autor, adaptado por FIAP (2017)

Essa associação pode ser exemplificada de uma forma um pouco mais detalhada (Figura “Exemplo de relacionamento entre médico e paciente (2)”).

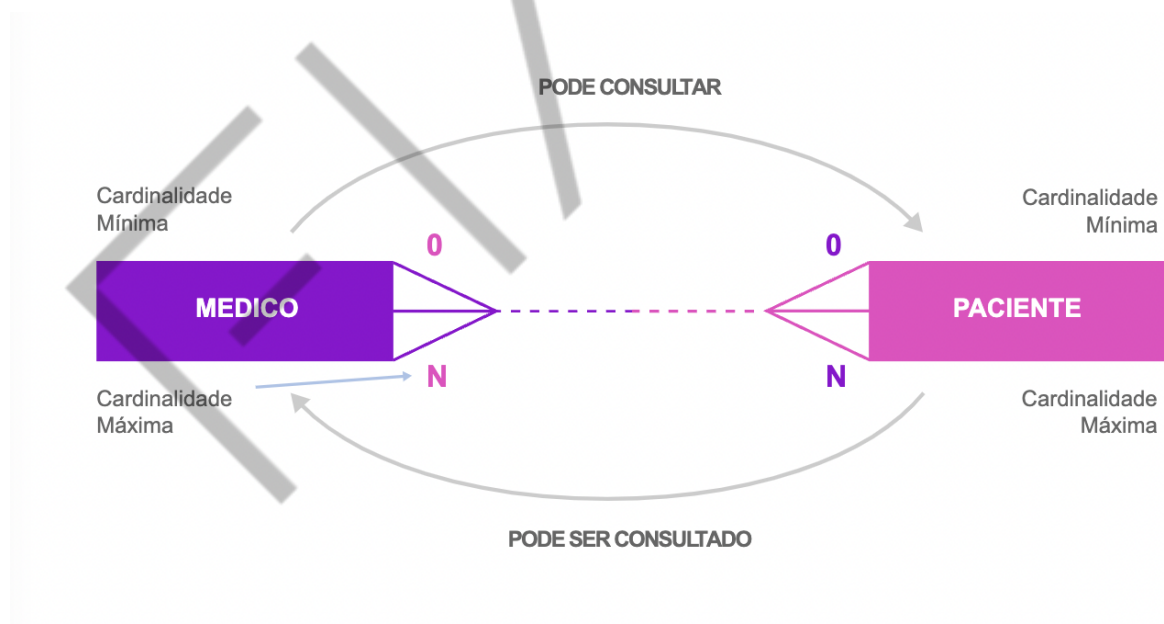


Figura 2 – Exemplo de relacionamento entre médico e paciente (2)  
Fonte: Elaborado pelo autor, adaptado por FIAP (2017)

Observe que temos um relacionamento com cardinalidade M:N. Quando pensamos em uma consulta médica, conseguimos encontrar atributos que caracterizam o relacionamento M:N. Por exemplo: uma consulta ocorre em uma data e hora e em uma determinada sala ou consultório.

Fique atento, data e hora da consulta não são características (atributos) de Médico, nem de Paciente, podemos dizer que a sala ou consultório também não caracterizam Médico ou Paciente, ou seja, essas características só existem quando uma consulta existir e, por esse motivo, são características (atributos) do relacionamento. Se a consulta não existir, essas características não existem.

A partir do momento em que encontramos os atributos que caracterizam o relacionamento e eles não pertencem unicamente a nenhuma das entidades relacionadas, observamos mais nitidamente a necessidade de uma nova entidade.

Neste exemplo, o nome mais adequado para a nova entidade é “CONSULTA”.

Quando transformamos o relacionamento em uma nova entidade, passamos a ter dois relacionamentos 1:N, um entre “MEDICO” e “CONSULTA” e outro entre “PACIENTE” e “CONSULTA” (Figura “Criação da entidade associativa consulta”).

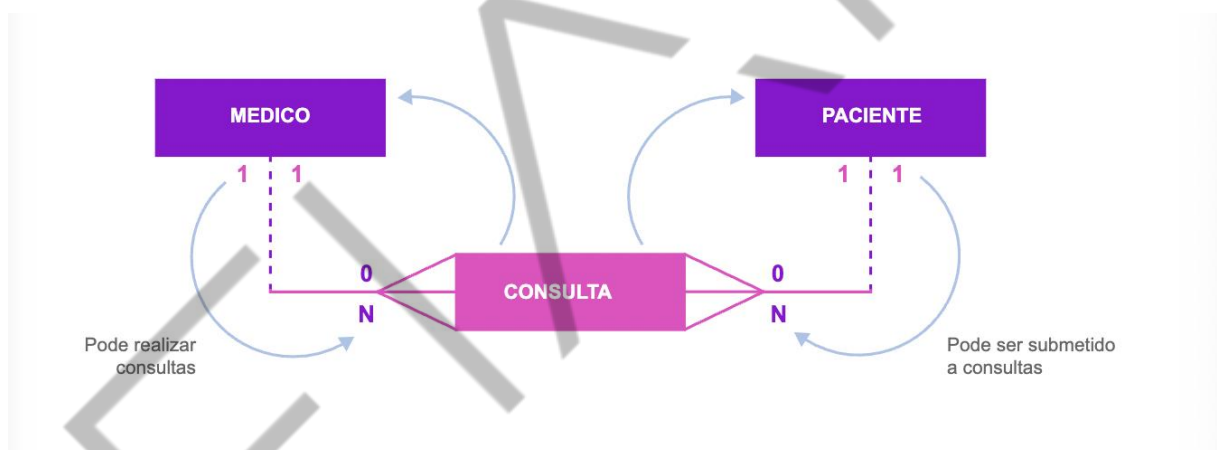


Figura 3 – Criação da entidade associativa consulta  
Fonte: Elaborado pelo autor, adaptado por FIAP (2017)

Características da entidade **CONSULTA** → { Data e hora, Sala ou consultório }.

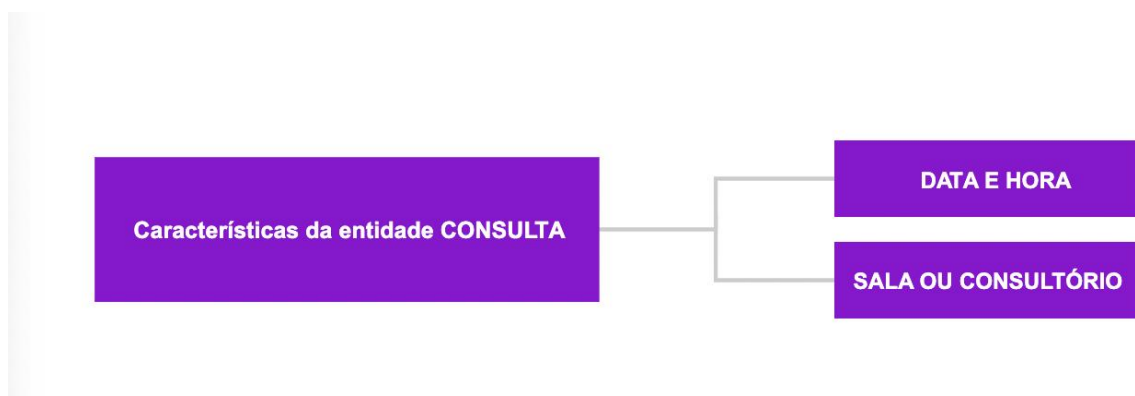




Figura 4 – Característica da entidade Consulta  
 Fonte: Elaborado pelo autor, adaptado por FIAP (2017)

Utilizando a representação gráfica por meio da ferramenta SQL Developer Data Modeler, a partir da notação da Engenharia da Informação, temos a representação do relacionamento M:N, entre “MEDICO” e “PACIENTE” (Figura “Relacionamento entre médico e paciente em um modelo lógico”).

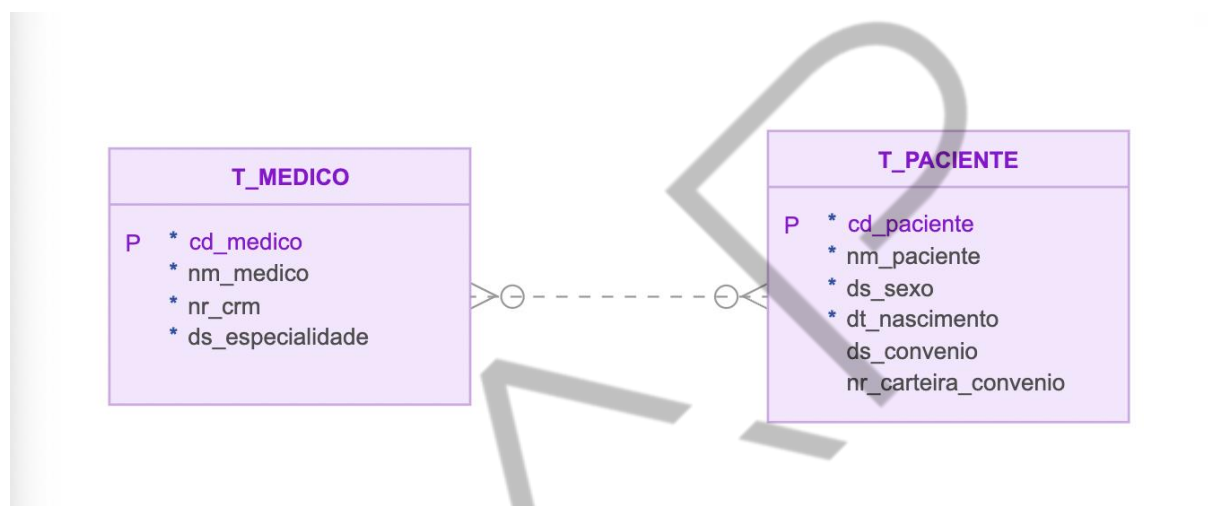


Figura 5 – Relacionamento entre médico e paciente em um modelo lógico  
 Fonte: Elaborado pela autora (2014)

Representação do relacionamento por meio da nova entidade “CONSULTA” (Figura “Relacionamento médico e paciente envolvendo a entidade associativa consulta”).

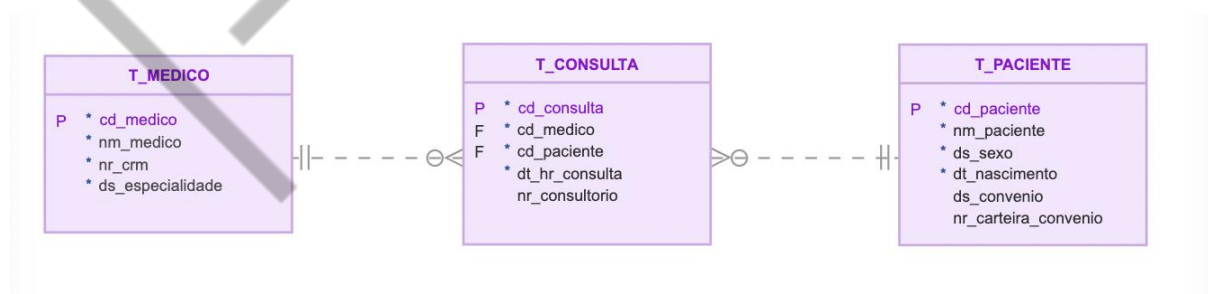


Figura 6 – Relacionamento médico e paciente envolvendo a entidade associativa consulta  
 Fonte: Elaborado pela autora (2014)

### 1.1.2 Regras ou soluções possíveis para determinar a chave primária em uma agregação

**1ª Solução:** Podemos criar uma chave primária composta pelas chaves estrangeiras, caso garantam unicidade das ocorrências, obtendo assim um relacionamento identificado.

**2ª Solução:** Podemos criar um novo atributo para ser a chave primária desta entidade. As chaves estrangeiras não são utilizadas como identificadores. O novo atributo será utilizado para garantir a unicidade das ocorrências.

**3ª Solução:** Podemos criar uma chave primária composta por uma das chaves estrangeiras, mais um atributo da própria entidade. Desde que juntos garantam a unicidade das ocorrências. Teríamos um relacionamento identificado de um dos lados. Considerando que a nova entidade é uma entidade FRACA, pois depende das outras entidades para existir, utilizamos, então, a mesma regra da chave primária da entidade FRACA.

No exemplo ilustrado acima, aplicamos a 2ª Solução para determinar a chave primária da nova entidade “CONSULTA”.

Vamos visualizar em termos de ocorrências para um melhor entendimento da solução proposta (Figura “Exemplo de ocorrências no relacionamento médico, consulta e paciente”).

**Entidade: PACIENTE**

cd_paciente	nm_paciente	ds_sexo	dt_nascimento	ds_convenio	nr_carteira_convenio
20151001	JOSÉ MARIA	M	03/07/1976	AMIL	100-00019878
20151002	ROSA MARIA	F	12/12/1960	ALLIANZ	110-09876453
20151003	MARIA DE LOURDES	F	17/09/1985	UNIMED	120-98764433
20151004	MANOEL PORTUGUÊS	M	23/11/1970	PORTO SEGURO	130-00876521
20151005	MARIA APARECIDA	F	27/12/1980	SUL AMERICA	140-00076542
20151006	ROBERTO DE SOUZA	M	19/08/1985	AMIL	100-00980765

**Entidade: MEDICO**

cd_medico	nm_medico	nr_crm	ds_especialidade
1234	JOÃO	32.004	CLÍNICO GERAL
1235	CARLOS	66.008	OFTALMOLOGIA
1236	MARIA	72.345	PEDIATRIA
1237	ROSA	11.234	ORTOPEDIA

**Entidade: CONSULTA**

cd_consulta	cd_medico	cd_paciente	dt_hr_consulta	nr_consultoria
201500001	1234	20151001	10/02/2015 10:00	1
201500002	1235	20151002	17/03/2015 09:00	2
201500003	1235	20151001	17/13/2015 10:00	3
201500004	1236	20151003	23/04/2015 10:00	5
201500005	1234	20151003	10/02/2015 10:45	
201500006	1236	20151001	23/04/2015 11:00	1
201500007	1235	20151004	17/03/2015 11:00	
201500008	1235	20151003	17/03/2015 13:00	4
201500009	1235	20151002	17/03/2015 18:00	2
201500010	1234	20151004	10/02/2015 11:15	
201500011	1236	20151006	23/04/2015 11:00	5

Figura 7 – Exemplo de ocorrências no relacionamento médico, consulta e paciente  
 Fonte: Elaborado pela autora (2014)

Fazendo a análise, observem que utilizamos a 2ª solução, criamos um novo atributo “CD\_CONSULTA” para a entidade “CONSULTA” para identificação única das ocorrências (chave primária).

Se utilizássemos a 1ª solução, teríamos repetição do par médico-paciente, pois existe a possibilidade de um médico atender um paciente mais de uma vez ao longo do tempo e até no mesmo dia.

Analisando a 3ª Solução, seria a utilização de uma das chaves estrangeiras mais um atributo da própria entidade, observamos que como o paciente e o médico podem se repetir, e os demais atributos, número da sala ou consultório são opcionais e a data não é um bom atributo para compor a chave, observamos que a 3ª Solução não é aplicável neste caso.

Observando o exemplo, vale ressaltar que para cada situação deve-se avaliar a melhor solução a ser aplicada de acordo com o contexto de negócio e as necessidades de armazenamento.

### 1.1.3 Evoluindo o exemplo de agregação

Suponha a situação: durante uma consulta (exemplo demonstrado acima), um Médico prescreve medicamentos para um Paciente.

Em cada consulta, o Médico pode prescrever medicamentos diferentes para o Paciente, ou até mesmo não prescrever nenhum medicamento.

Mas, para existir uma prescrição (receita que o Médico nos fornece durante uma consulta), o paciente deve obrigatoriamente passar por uma consulta médica.

Portanto, a prescrição seria um relacionamento entre uma entidade Medicamento e o relacionamento consulta.

Cada consulta pode prescrever vários medicamentos ou nenhum. Cada medicamento pode ser prescrito em várias consultas ou nenhuma.

Podemos representar a associação entre médico (Figura “Relacionamento entre consulta e medicamento”).

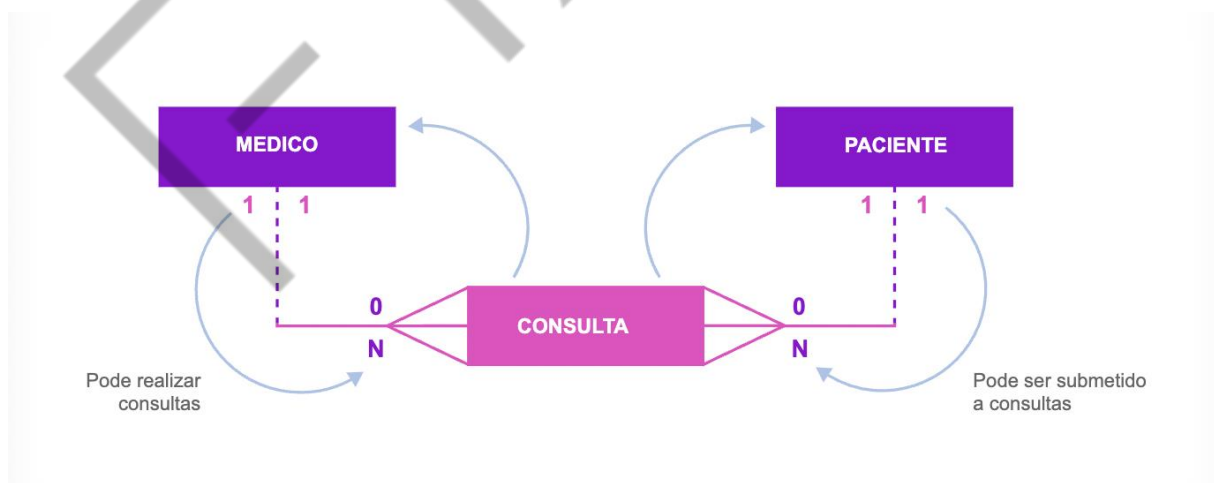


Figura 8 – Relacionamento entre consulta e medicamento  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Exemplificando de uma forma um pouco mais detalhada (Figura “Relacionamento entre consulta e medicamento (2)”).

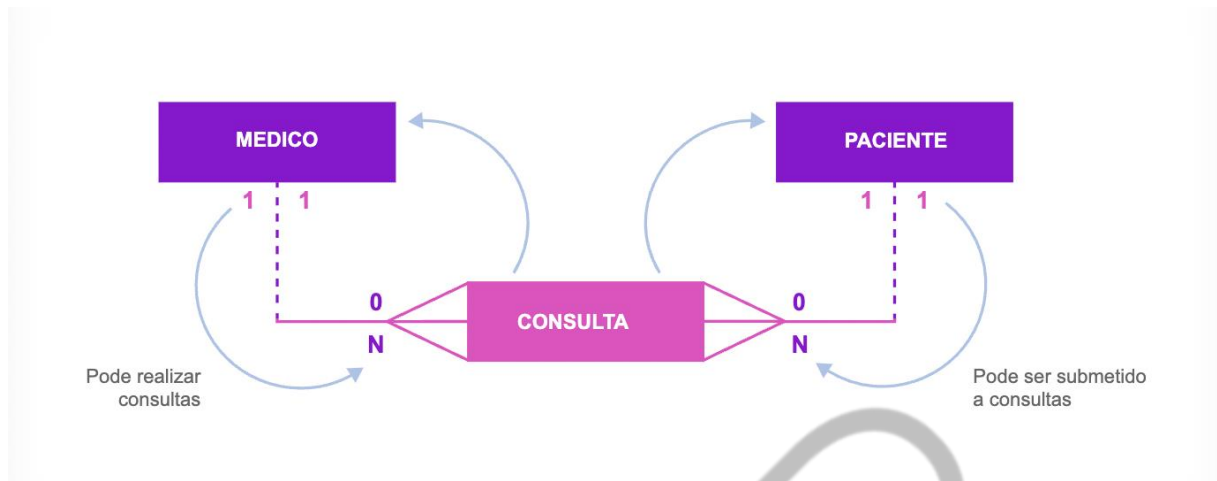


Figura 9 – Relacionamento entre consulta e medicamento (2)  
 Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Observe que temos um relacionamento com cardinalidade M:N.

Veja que, quando pensamos em uma prescrição de medicamentos durante uma consulta, conseguimos encontrar atributos que caracterizam o relacionamento M:N. Por exemplo, uma prescrição contém posologia, via, orientações de uso, além, é claro, do medicamento, nome do médico, CRM e data da prescrição (encontrado na descrição da entidade Medicamento, Médico e Consulta, respectivamente, não é necessário repetir esses atributos).

Exemplificando a estrutura mínima da prescrição, temos: Posologia (dosagem certa de um medicamento) = 1 comprimido de 50 mg Via = Oral; orientações de uso = a cada 8 horas após as refeições.

Analisando esse exemplo, observamos que posologia, via e orientações de uso são atributos que só existem quando uma prescrição existir, portanto, não caracteriza Consulta e nem caracteriza Medicamento. Podemos dizer que são atributos do relacionamento. Se Prescrição não existir, os atributos citados não se aplicam às demais relações existentes.

A partir do momento em que encontramos os atributos que caracterizam o relacionamento, observamos mais nitidamente a necessidade de uma nova entidade.

Neste exemplo, o nome mais adequado para a nova entidade é “PRESCRICAO”. Mas por que não “RECEITA”? Porque receita representa uma espécie de relatório que contém todos os medicamentos prescritos por um médico durante uma consulta. Pensando em armazenamento, precisamos guardar cada um dos medicamentos prescritos em uma determinada consulta.

Quando transformamos o relacionamento em uma nova entidade, passamos a ter dois relacionamentos 1:N, um entre “CONSULTA” e “PRESCRICAO” e outro entre “MEDICAMENTO” e “PRESCRICAO” (Figura “Criação da entidade associativa prescrição entre consulta e medicamento”).

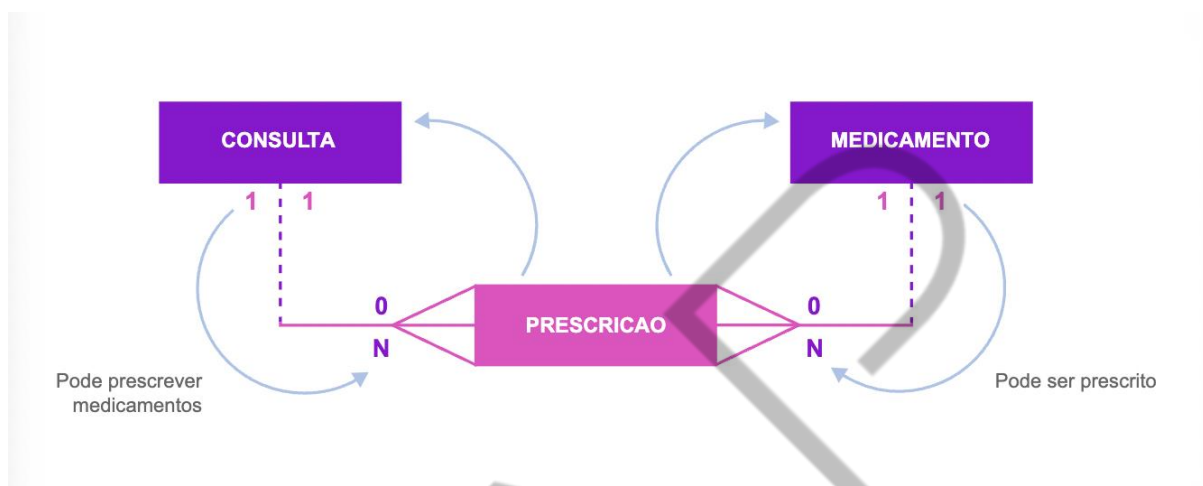


Figura 10 – Criação da entidade associativa prescrição entre consulta e medicamento  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Características da entidade **PRESCRICAO** → { Posologia, Via, Orientações de uso} (Figura “Característica da entidade PRESCRICAO”).

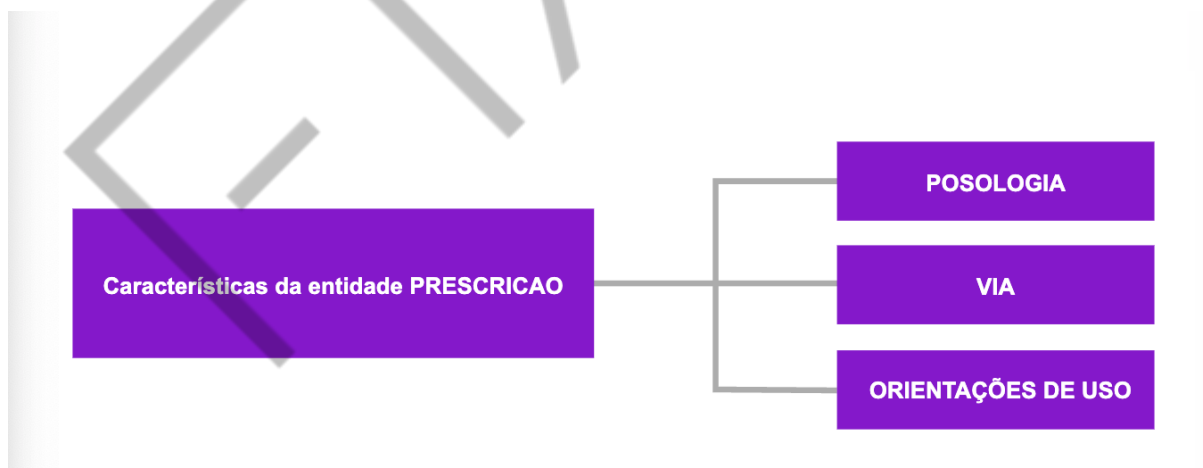


Figura 11 – Característica da entidade PRESCRICAO  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Utilizando a representação gráfica por meio da ferramenta SQL Developer Data Modeler, a partir da notação da Engenharia da Informação:

Representação do relacionamento M:N, entre “CONSULTA” e “MEDICAMENTO” (Figura “Relacionamento entre consulta e medicamento no Modelo Lógico”).

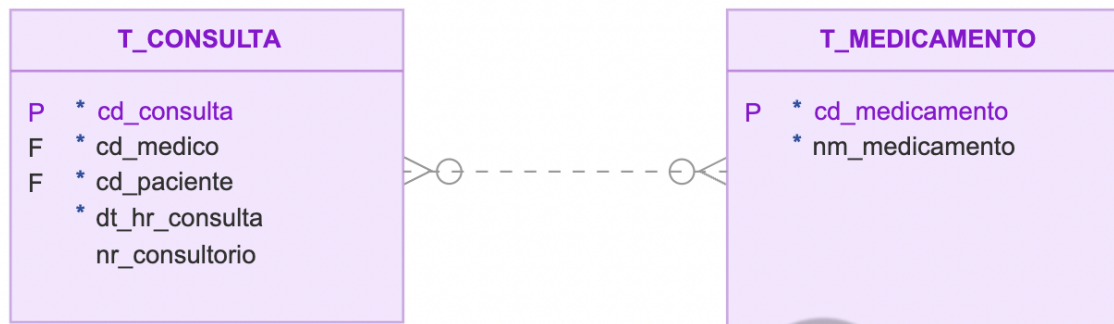


Figura 12 – Relacionamento entre consulta e medicamento no Modelo Lógico  
 Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Representação do relacionamento por meio da nova entidade “PRESCRICAO” (Figura “Relacionamento entre consulta, prescrição e medicamento no Modelo Lógico”).

#### SQL Data Modeler – Notação de Barker:

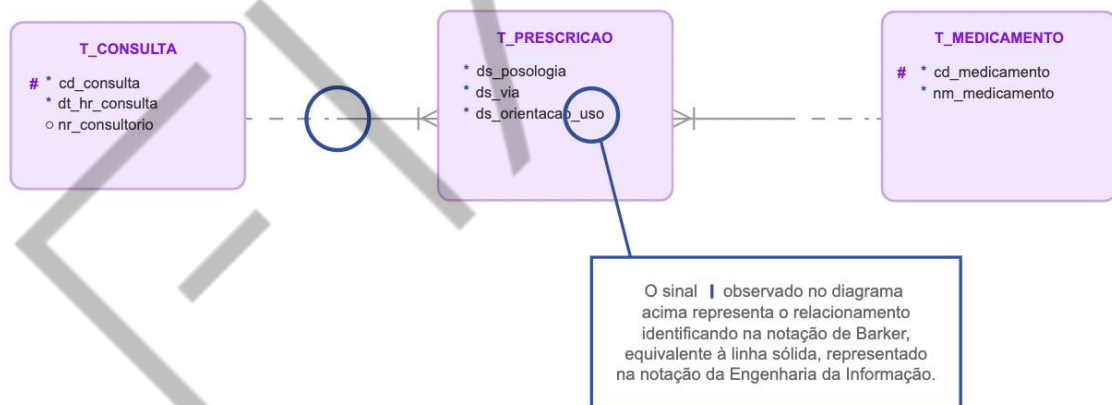


Figura 13 – Relacionamento entre consulta, prescrição e medicamento no Modelo Lógico  
 Fonte: Elaborado pela autora, adaptado por FIAP (2017)

#### SQL Data Modeler – Notação da Engenharia da Informação:



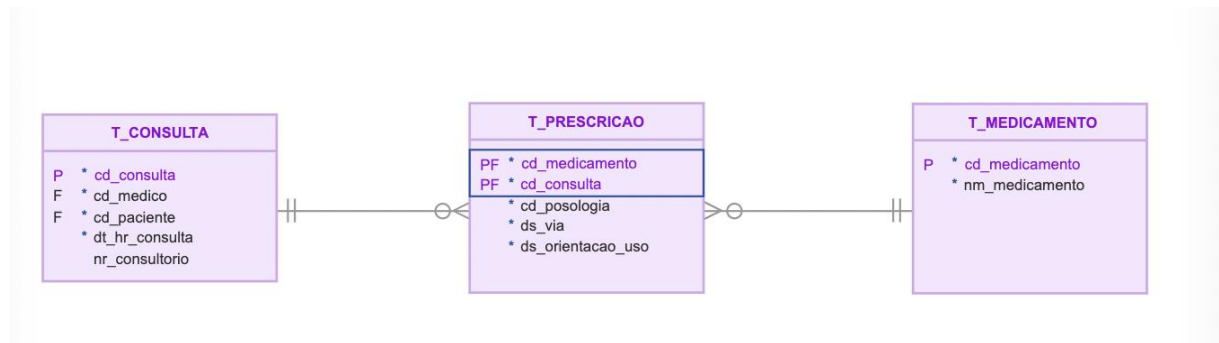


Figura 14 – Classe associativa prescrição com uma chave composta

Fonte: Elaborado pela autora, adaptado por FIAP (2017)

No exemplo ilustrado acima, aplicamos a 1ª Solução para determinar a chave primária da nova entidade “PRESCRICAO” (Figura “Classe associativa prescrição com uma chave composta”).

Vamos visualizar em termos de ocorrências para um melhor entendimento da solução proposta (Figura “Exemplos de ocorrências das futuras tabelas consulta, prescrição e medicamento”).

cd_consulta	cd_medico	cd_paciente	dt_hr_consulta	nr_consultoria
201500001	1234	20151001	10/02/2015 10:00	1
201500002	1235	20151002	17/03/2015 09:00	2
201500003	1235	20151001	17/13/2015 10:00	3
201500004	1236	20151003	23/04/2015 10:00	5
201500005	1234	20151003	10/02/2015 10:45	
120150006	1236	20151001	23/04/2015 11:00	1
201500007	1235	20151004	17/03/2015 11:00	
201500008	1235	20151003	17/03/2015 13:00	4
201500009	1235	20151002	17/03/2015 18:00	2
201500010	1234	20151004	10/02/2015 11:15	
201500011	1236	20151006	23/04/2015 11:00	5

cd_medicamento	nm_medicamento
100100	Novalgina
100101	Rinossoro
100102	Advil
100103	Tilenol
100104	Omeprazol
100105	Dorflex

cd_consulta	cd_medicamento	ds_posologia	ds_via	ds_orientacao_uso
201500001	100100	40 gotas	oral	A cada 8h
201500001	100101	25 gotas	nasal	A cada 8h
201500001	100102	1 comprimido	oral	A cada 6h
201500003	100102	1 comprimido	oral	A cada 8h
201500003	100103	1 comprimido	oral	A cada 6h
120150007	100102	1 comprimido	oral	A cada 8h
201500007	100104	1 comprimido	oral	A cada 8h

Chave primária composta pelos atributos  
 “cd\_consulta” e “cd\_medicamento”.  
 Os dois atributos garantem a identificação única

Figura 15 – Exemplos de ocorrências das futuras tabelas consulta, prescrição e medicamento

Fonte: Elaborado pela autora, adaptado por FIAP (2017)



Fazendo a análise, utilizamos a **1ª solução**, as chaves estrangeiras para compor a chave primária da entidade “PRESCRICAO”, podemos observar que um medicamento não é prescrito mais de uma vez para uma mesma consulta. Identificando de forma única cada uma das ocorrências.

Temos, então, uma chave primária composta pelas chaves estrangeiras na entidade “PRESCRICAO”. Quando isso ocorre, temos um **relacionamento identificado**.

O Relacionamento identificado recebe esse nome em função das chaves estrangeiras serem utilizadas como chave primária na nova entidade, ou seja, são identificadores na entidade “PRESCRICAO”.

Vale ressaltar que quando escolhemos a 1ª Solução (chave composta) vale uma análise da solução proposta como um todo, pois qualquer outra entidade que se relacionar com Prescrição terá, como chave estrangeira, uma chave composta por dois atributos. O cuidado é em relação aos demais relacionamentos que podem existir, tornando as chaves primárias ou estrangeiras longas demais. Nessas situações, vale a utilização da 2ª Solução, que é a criação de um novo atributo como chave primária.

Como ponto de atenção, podemos dizer que não é boa prática criar chaves primárias compostas com mais de dois atributos, a chave fica longa, ocupa espaço em disco, o SGBD tem um custo maior para ordenação e busca dos dados por meio da chave.

#### 1.1.4 Um pouco mais de atributos do relacionamento

Vamos abordar outro exemplo, no qual teremos atributos do relacionamento. No exemplo a seguir, podemos dizer que um engenheiro pode exercer funções diferentes em diferentes projetos nos quais atuar, portanto, na representação do relacionamento “ATUACAO”, podemos utilizar um atributo que determinará a função que o engenheiro exerce dentro de um projeto.

O atributo função não pode ser considerado do “ENGENHEIRO”, uma vez que ele pode atuar em diversos projetos; mas também não pode ser um atributo do “PROJETO”, uma vez que em um projeto podem atuar vários engenheiros com

funções diferentes (Figura “Modelo conceitual do relacionamento entre engenheiro e projeto”).

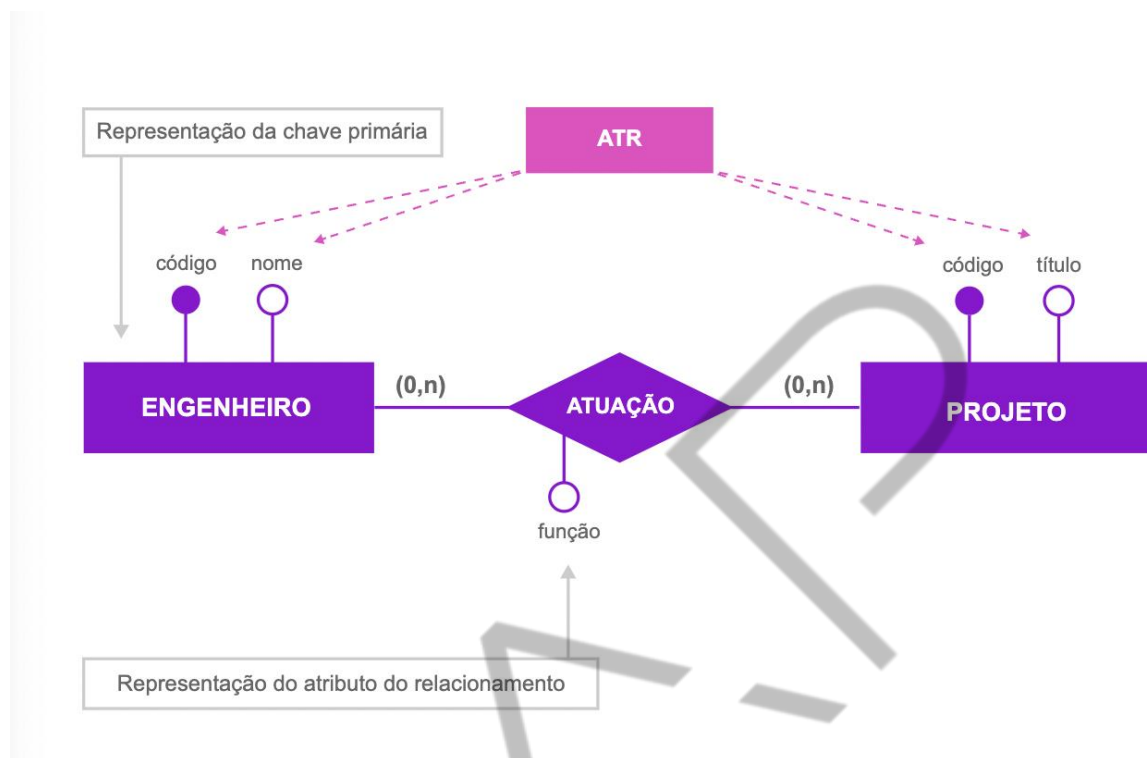


Figura 16 – Modelo conceitual do relacionamento entre engenheiro e projeto  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Representação do relacionamento M:N, entre “ENGENHEIRO” e “PROJETO”. Utilizando a notação da Engenharia da Informação (Figura “Modelo lógico do relacionamento entre engenheiro e projeto”).

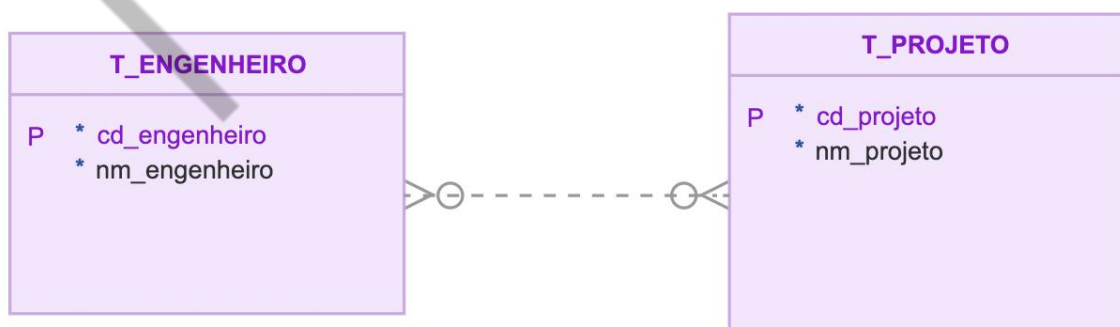


Figura 17 – Modelo lógico do relacionamento entre engenheiro e projeto  
Fonte: Elaborado pela autora (2014)

O atributo função (**atributo do relacionamento**) gera uma nova entidade, que podemos chamar “ENGENHEIRO\_PROJETO” ou “ATUACAO” ou outro nome que for pertinente, de acordo com a necessidade de negócio. Podemos, inclusive, inserir outros atributos que poderão facilitar análises futuras, como início e término da função no projeto, que também são atributos que caracterizam a associação M:N entre ENGENHEIRO e PROJETO.

Dessa forma, poderíamos ter: representação do relacionamento por meio da nova entidade “ATUACAO”.

### SQL Data Modeler – Notação de Barker:

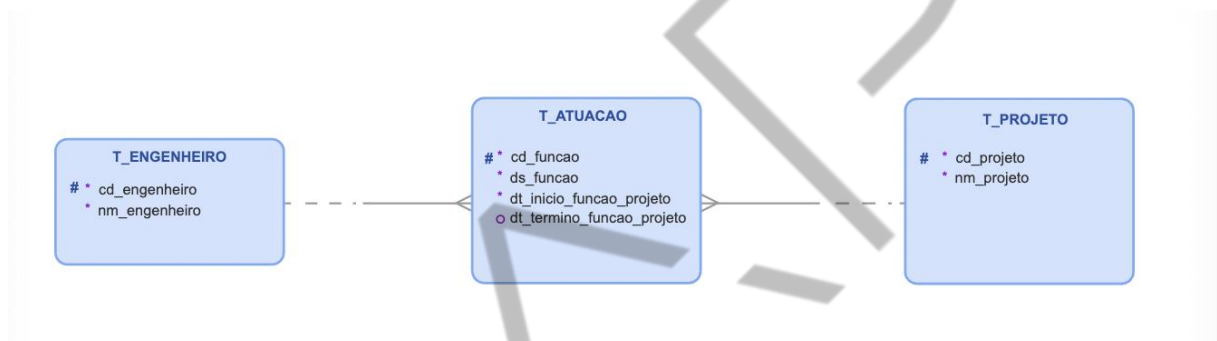


Figura 18 – Criação da entidade associativa atuação, entre as entidades engenheiro e projeto  
Fonte: Elaborado pela autora (2014)

### SQL Data Modeler – Notação da Engenharia da Informação:

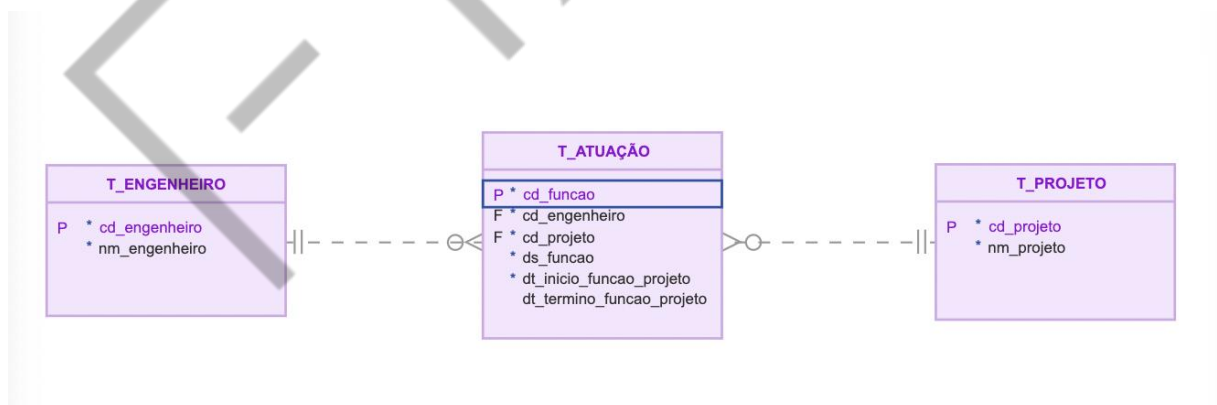


Figura 19 – Abordagem criando uma chave para a entidade associativa atuação  
Fonte: Elaborado pela autora (2014)

Pensando na chave primária da nova entidade “ATUACAO”, **podemos aplicar a 2ª Solução**, pois um mesmo engenheiro poderia entrar e sair de um projeto várias vezes, de acordo com a necessidade, e até executar diferentes funções dentro de um mesmo projeto.

Foi, então, criado o atributo “CD\_FUNCAO” (Figura “Abordagem criando uma chave para a entidade associativa atuação”).

### 1.1.5 Exemplificando outra situação na qual temos entidade associativa

É possível que, em alguns relacionamentos M:N, não seja possível encontrar atributos do relacionamento em função das características de negócio.

Quanto isso ocorre, a entidade associativa terá apenas as chaves estrangeiras. Você pode se perguntar, em um primeiro momento, se isso é possível. A resposta é SIM, é possível, a nova entidade funcionará como ligação entre as entidades associadas para resolver o problema que temos em relação à implementação o relacionamento M:N.

Quanto à chave primária dessa nova entidade, ela segue as mesmas regras descritas acima. Neste capítulo, devemos analisar a melhor solução de acordo com o contexto de negócio. Podendo até surgir a necessidade da criação de um novo atributo, caso a utilização das chaves estrangeiras, formando uma chave primária composta, não garanta a unicidade de cada ocorrência da nova entidade.

Vejamos um exemplo, em que seja necessário saber quais professores estão habilitados a lecionar “tal” disciplina.

Exemplo: Um professor pode lecionar uma ou mais disciplinas desde que tenha habilitação para ministrá-las. Cada disciplina deve ser lecionada por um ou mais professores que tenham habilitação para ministrá-las.

Representação do relacionamento muitos para muitos.

**SQL Data Modeler – Notação da Engenharia da Informação:**

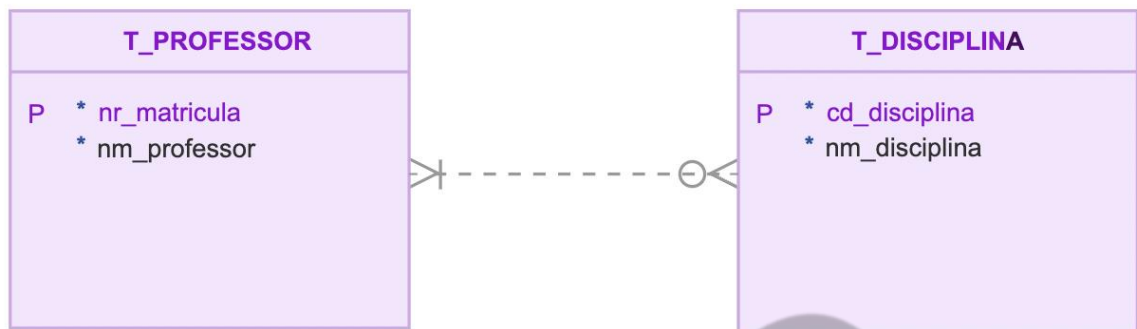


Figura 20 – Modelo lógico das entidades professor e disciplina  
Fonte: Elaborado pela autora (2014)

Representação do relacionamento por meio da nova entidade “PROFESSOR\_DISCIPLINA”.

### SQL Data Modeler – Notação da Engenharia da Informação:

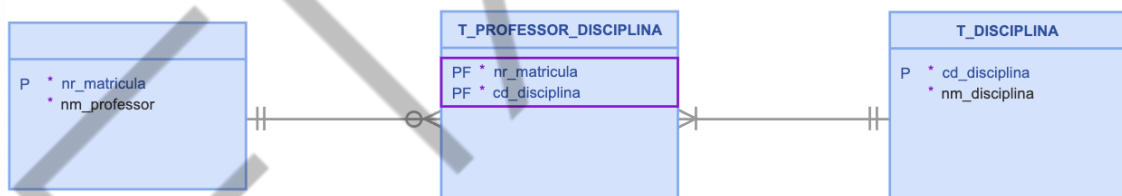


Figura 21 – Criação da entidade associativa professor\_disciplina  
Fonte: Elaborado pela autora (2014)

Observe que a chave primária é composta pelas estrangeiras, uma vez que um professor esteja apto a cada disciplina uma única vez ao longo do tempo, ou seja, não iremos cadastrar um professor habilitado em matemática duas, três, quatro vezes. Associaremos o professor à disciplina a que ele está habilitado uma única vez. Portanto, a chave primária composta garantirá a unicidade das ocorrências da entidade “PROFESSOR\_DISCIPLINA”.

## 1.2 Sobre herança

Podemos dizer que um modelo de dados pode e deve passar por um processo de refinamento. Observamos algumas situações particulares, que chamamos de extensões, cuja função é de possibilitar a correção dessas particularidades para que o modelo de dados possa ser posteriormente implementado de forma física, por meio de um SGBD.

A herança é uma forma de ajustar a implementação quando temos entidades que possuam as mesmas características (atributos) e desempenhem as mesmas ações (relacionamentos).

Aplicamos o conceito de herança que foi abordado em capítulos anteriores, por meio da generalização e especialização, que são conceitos utilizados para representar entidades que possuem os mesmos atributos e que podem ser categorizadas.

A especialização/generalização promove uma maior flexibilidade ao modelo, permitindo:

- Definir um conjunto de subclasses de um tipo de entidade.
- Definir atributos específicos para cada subclasse.
- Definir relacionamentos específicos entre as subclasses e outras subclasses, ou outras entidades.

Especialização é o conceito que permite atribuir propriedades particulares a um subconjunto das ocorrências (especializadas) de uma entidade genérica.

Os atributos inseridos na entidade especializada não se aplicam a todas as entidades. A entidade que possuir propriedades genéricas será chamada SUPERCLASSE ou ENTIDADE GENÉRICA.

A entidade que possuir propriedades particulares será chamada de SUBCLASSE ou ENTIDADE ESPECÍFICA/ESPECIALIZADA.

Generalização é o conceito que permite unir dois ou mais conjuntos de subclasses (entidades com propriedades particulares), produzindo uma entidade com um conjunto de propriedades genéricas (Superclasse).

Os atributos inseridos na entidade genérica são comuns a dois ou mais conjuntos de entidades, assim, a herança de Atributos é uma consequência desse processo de especialização/generalização.

As subclasses (entidades especializadas) herdam os atributos da superclasse (entidade genérica), ou seja, cada ocorrência da entidade especializada possui, além de suas próprias propriedades (atributos e relacionamentos), as propriedades da ocorrência da entidade genérica correspondente.

### 1.2.1 Exemplificando especialização, generalização e herança

#### Exemplo 1:

Suponha esta situação: temos um conjunto de dados que caracterizam clientes que são pessoas físicas e clientes que são pessoas jurídicas (Figura “Exemplos de ocorrências para pessoas físicas e jurídicas”).

PESSOA FÍSICA								
CODIGO	NOME	RG	CPF	DATA NASCIMENTO	SEXO	EMAIL	ENDEREÇO	TELEFONE
1	JOAO	123456789-0	111111 1 1-11	12/03/1984	M	joao@joao.com.br	R. X, 25	1234-1234
2	JOSE	23456789-1	22222222-22	24/04/1996	M	jose@jose.com.br	R. Y, 14	2345-2345
3	ROSA	345678901-2	33333333-33	21/05/1983	F	rosa@rosa.com.br	AV, XYZ, 57	3456-3456
4	JOANA	456789012-3	44444444-44	17/10/1950	F	joana@joana.cof.br	TRAV. ABC, 3	1212-1212
5	MARIA	567890123-4	55555555-55	14/07/1987	F	maria@maria.cof.br	PÇ TATU, 25	3232-4444

PESSOA JURÍDICA								
CODIGO	RAZÃO SOCIAL	NOME FANTASIA	CNPJ	INSC. ESTADUAL	RAMO ATIVIDADE	EMAIL	ENDEREÇO	TELEFONE
1	Companhia Bras. Distribuição	EXTRA SUPERMERCADO	125125125/0001-00	12345678-12	ALIMENTICIO	extra@extra.com.br	R. XDS, 28	8234-1234
2	Carrefour Com. Indústria Ltda.	CARREFOUR	222111222/0001-12	23456789-23	ALIMENTICIO	carrefour@carrefour.com.br	R. VAC, 15	5345-2345
3	Lojas Americanas S/A	AMERICANAS	321321321/0001-23	45678901-11	COMERCIO	americanas@americanas.com.br	AV, ZXC, 58	2456-3456
4	CeC Casa e Construção Ltda.	C&C	452567890/0001-33	34212345-22	MATERIAL CONSTRUÇÃO	cec@cec.com.br	TRAV. CDE, 32	2212-1212
5	Saint-Gobain Distribuição Brasil Ltda.	Telhanorte	234627890/0001-99	21213454-01	MATERIAL CONSTRUÇÃO	telhanorte@telhanorte.com.br	PÇ XV, 24	4232-4444

Figura 22 – Exemplos de ocorrências para pessoas físicas e jurídicas  
Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

Observamos que há atributos comuns aos dois conjuntos de dados, por exemplo, pessoa física e pessoa jurídica possuem: código, nome, e-mail, endereço e telefone.

Por um lado, existem atributos que são específicos a “**PESSOA FÍSICA**” como, por exemplo: RG, CPF, data de nascimento e sexo. Por outro lado, existem atributos específicos a “**PESSOA JURÍDICA**” como, por exemplo: razão social, CNPJ, inscrição estadual e ramo de atividade.

Poderíamos, então, separar esses grupos de dados em:

- PESSOA - teríamos os atributos comuns.
- PESSOA FÍSICA - teríamos os atributos específicos dos clientes pessoa física.
- PESSOA JURÍDICA - teríamos os atributos específicos dos clientes pessoa jurídica.

Podemos representar a herança da seguinte maneira:

**Notação de Peter Chen:**

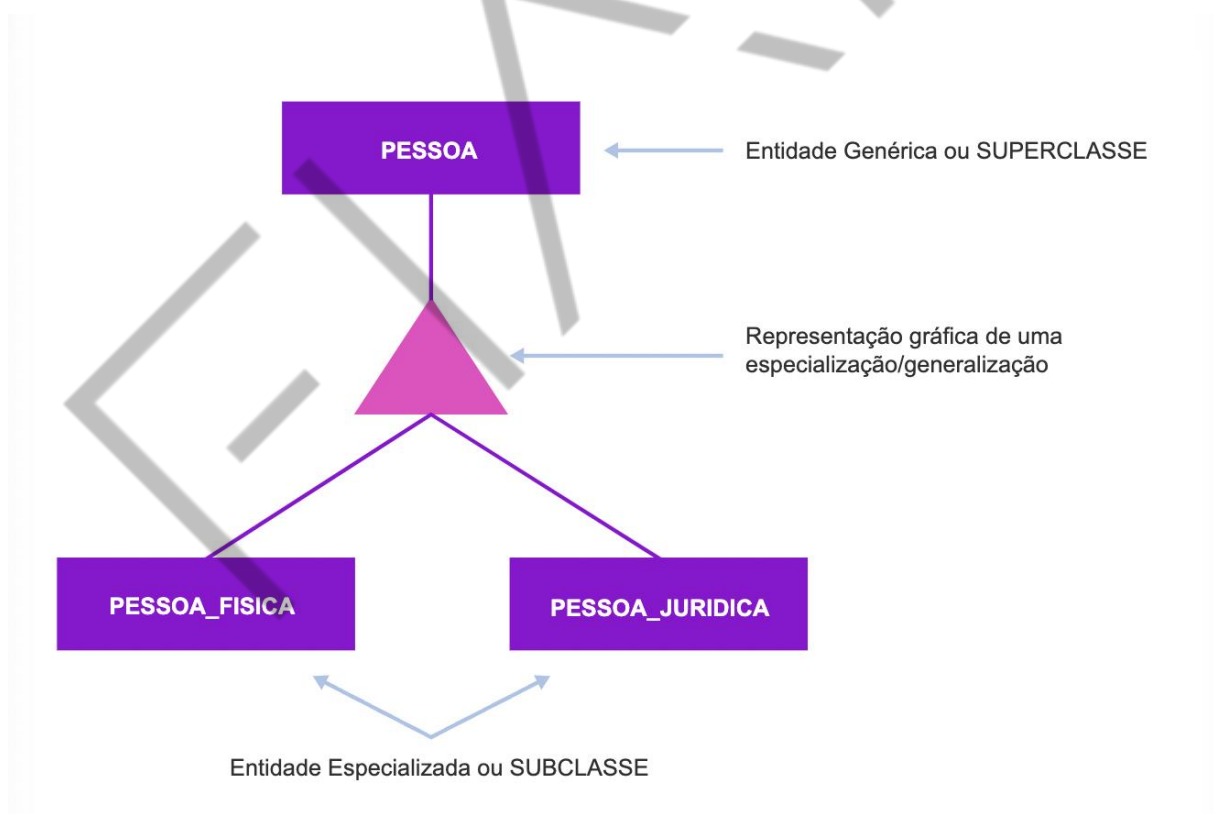


Figura 23 – Relacionamento entre pessoa, pessoa física e pessoa jurídica  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

**SQL Data Modeler – Notação de Barker:**



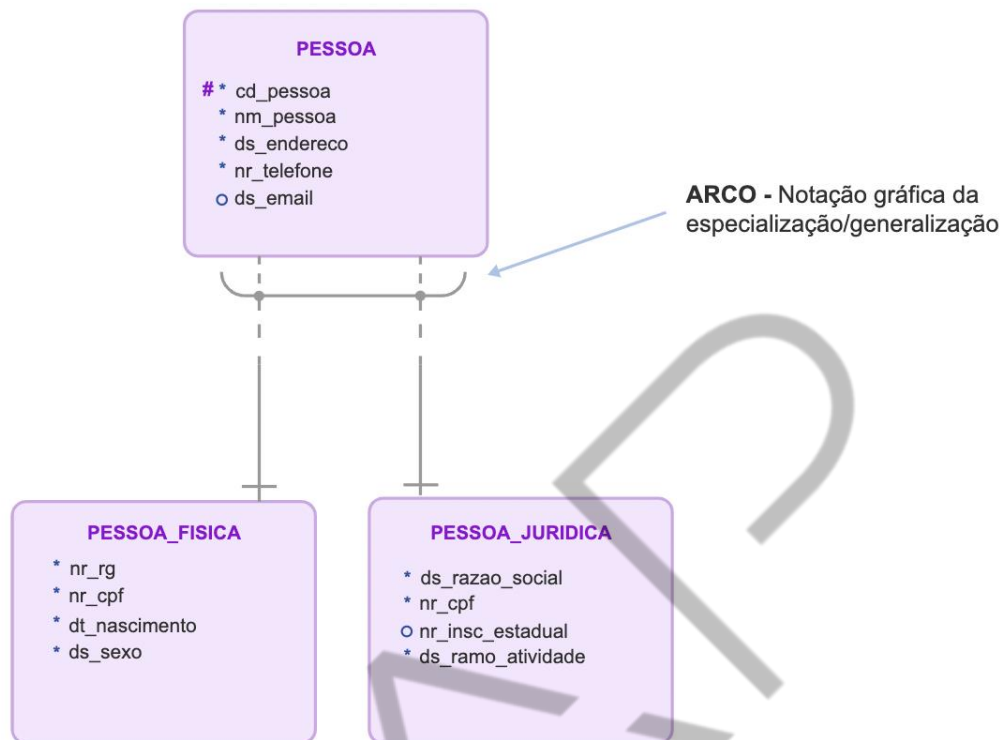


Figura 24 – Modelo lógico utilizando um arco, notação de Barker  
Fonte: Elaborado pela autora, adaptador por FIAP (2017)

### SQL Data Modeler – Notação da Engenharia da Informação:

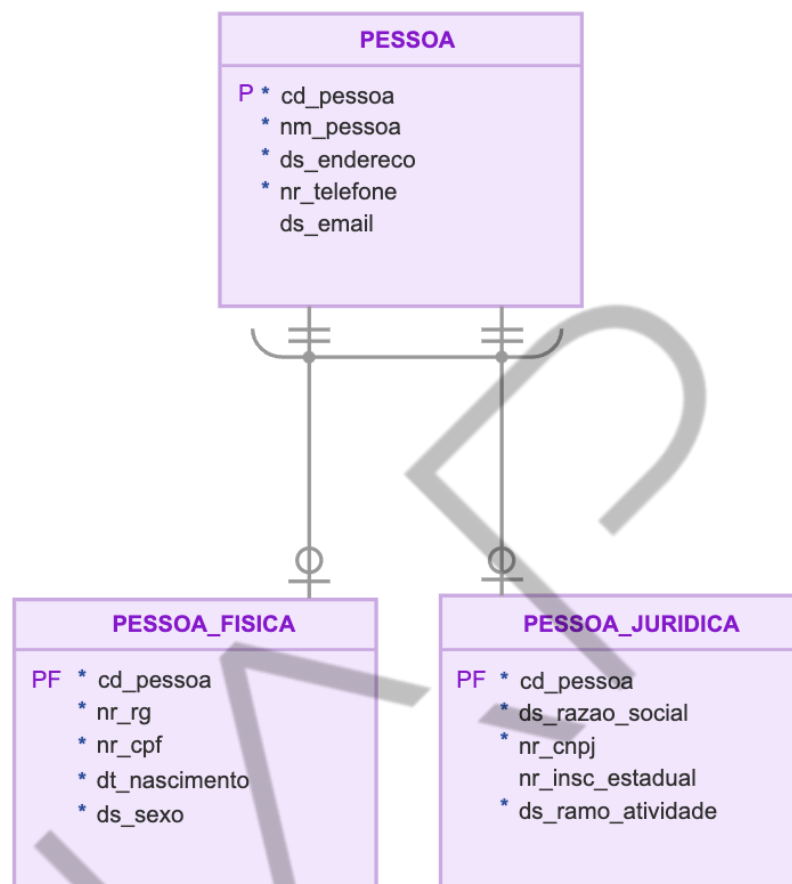


Figura 25 – Modelo lógico utilizando um arco, notação de Engenharia da Informação  
 Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

Quando utilizamos o conceito de especialização/generalização temos a ideia de **herança de atributos**. O que significa que cada ocorrência das entidades (entidades especializadas) “PESSOA\_FISICA” ou “PESSOA\_JURIDICA”, **além dos seus próprios atributos, possuirá também os atributos (herança)** da entidade genérica “PESSOA”.

**Exemplo:** A entidade “PESSOA\_FISICA” possuirá seus **próprios atributos** (atributos específicos: RG, CPF, data nascimento e sexo) e, também, os **atributos da entidade genérica** (atributos comuns: código, nome, endereço, telefone e e-mail).

Observando as representações gráficas, teremos um **relacionamento do tipo 1:1** entre a entidade “PESSOA” e “PESSOA\_FISICA”; “PESSOA” e

“**PESSOA\_JURIDICA**” (Figuras “Relacionamento entre pessoa, pessoa física e pessoa jurídica”, “Modelo lógico utilizando um arco, notação de Barker” e “Modelo lógico utilizando um arco, notação de Engenharia da Informação”).

### Interpretação:

Cada pessoa pode ser uma única pessoa física.

Cada pessoa pode ser uma única pessoa jurídica.

É importante destacar que, no caso na especialização, as entidades representam um mesmo objeto, porém cada entidade especializada desempenha um papel específico dentro do modelo de dados.

### Exemplo 2:

Suponha a situação abaixo: temos um conjunto de dados que caracterizam **os funcionários de uma empresa** (Figura “Exemplos de ocorrências entre funcionário, engenheiro, secretária e médico”).

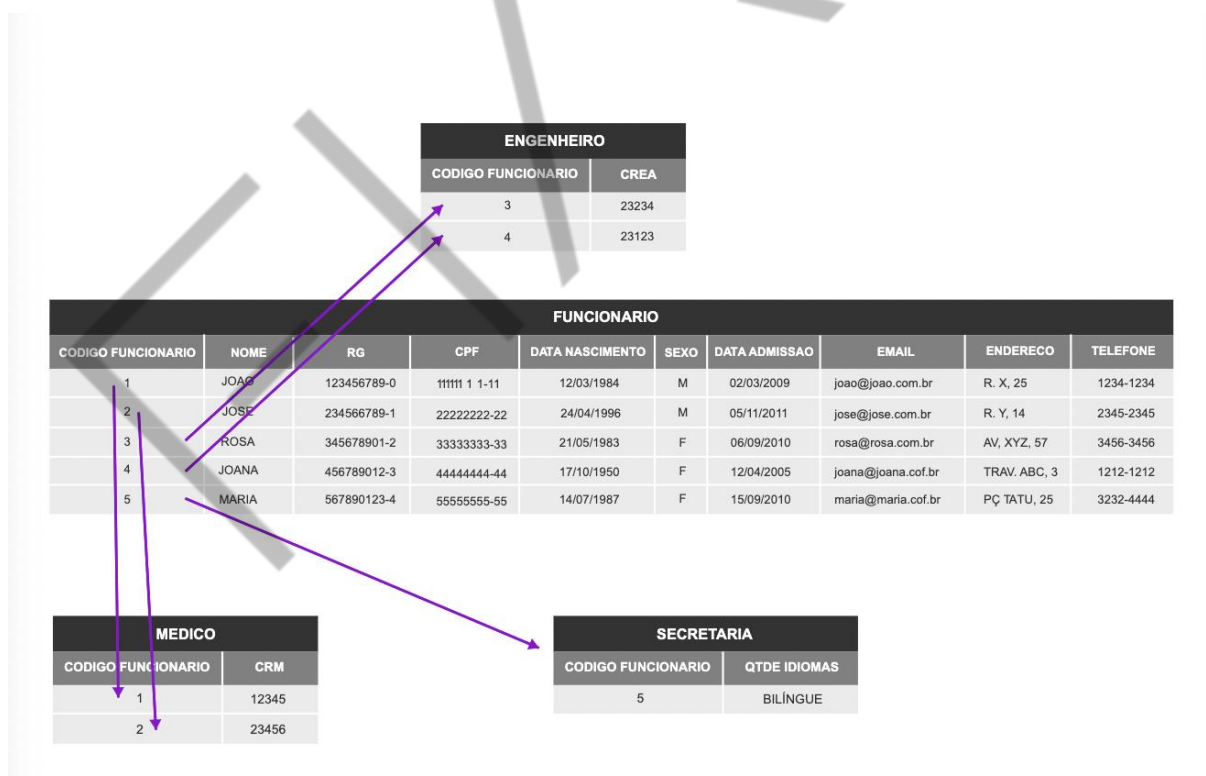


Figura 26 – Exemplos de ocorrências entre funcionário, engenheiro, secretária e médico  
Fonte: Elaborado pela autora (2017), adaptado por FIAP (2017)

Observamos que **há dados que caracterizam alguns funcionários da empresa**, por exemplo: um médico possui CRM, um engenheiro possui CREA, uma

secretária pode falar várias línguas. No entanto, note que a maior parte dos atributos é comum a todos os tipos de funcionários.

Notamos também que os funcionários: Médico, Engenheiro e Secretária possuem atributos específicos. Assim sendo, faz sentido especializar a entidade “FUNCIONARIO” a partir da categoria funcional, se esses funcionários possuírem atributos ou relacionamentos próprios.

Caso não exista, dentro do contexto de negócio, tal necessidade, podemos criar uma única entidade com todos os atributos (comuns e específicos) ou ainda criar três entidades e replicar seus atributos comuns. Portanto, poderíamos implementar essa situação no modelo de dados, a partir da especialização/ generalização.

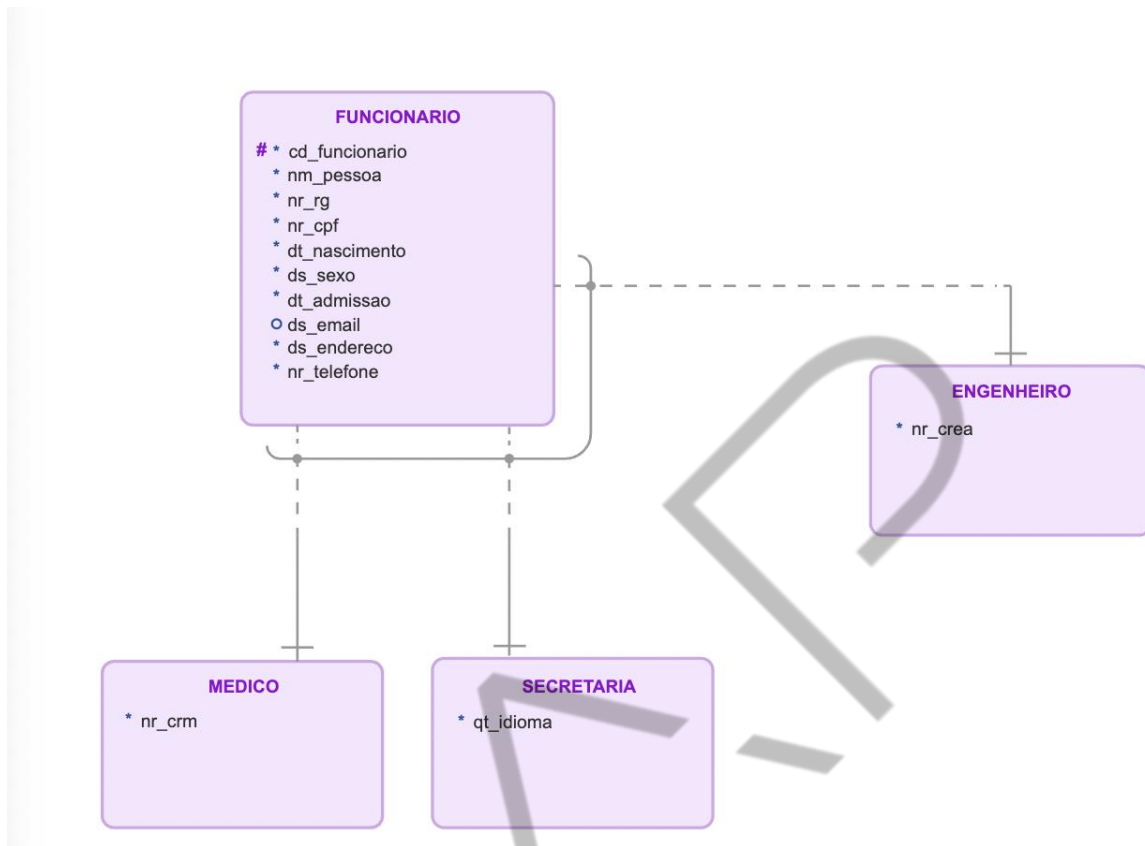
**SQL Data Modeler – Notação de Barker:**

Figura 27 – Modelo lógico envolvendo funcionário, engenheiro, secretária e médico, notação de Barker

Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

**SQL Data Modeler – Notação da Engenharia da Informação**

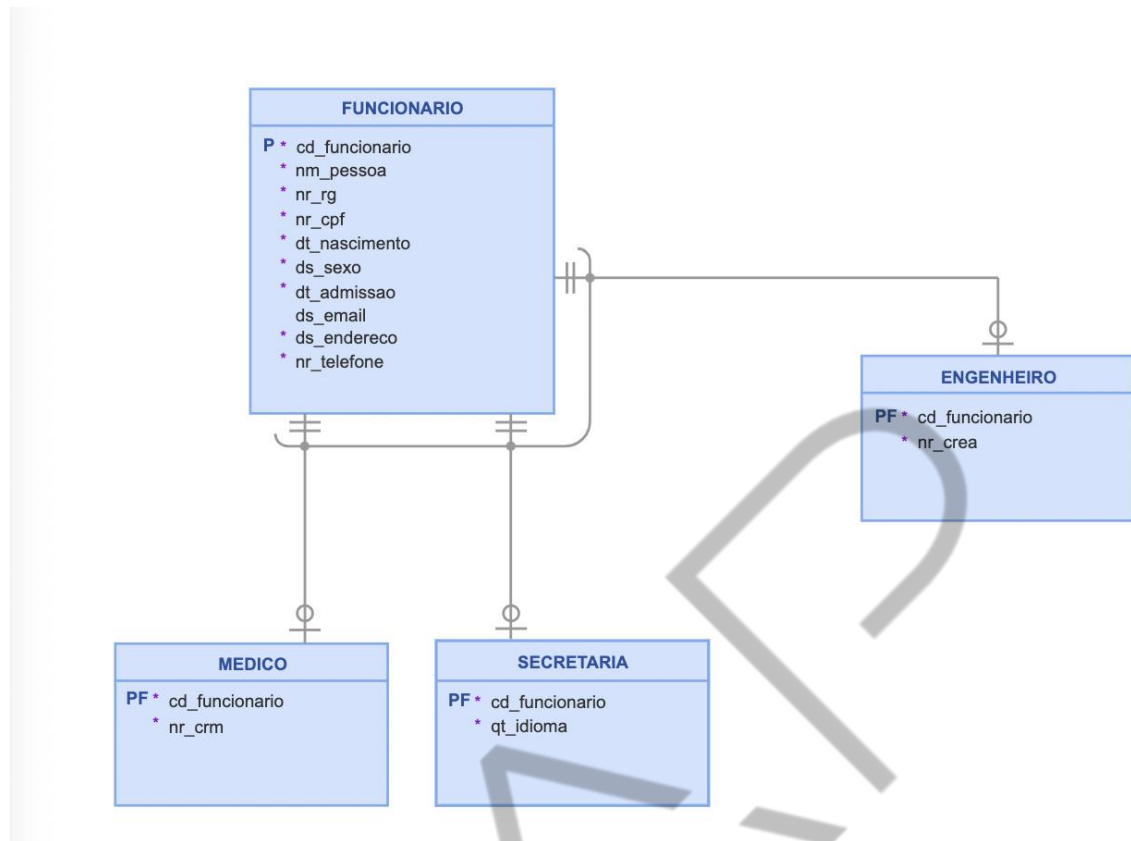


Figura 28 – Modelo lógico envolvendo funcionário, engenheiro, secretária e médico, notação de Engenharia da Informação

Fonte: Elaborado pela autora (2017), adaptado por FIAP (2017)

**Exemplo:** utilizando uma única entidade com todos os atributos, para a situação proposta no exemplo 2.

**SQL Data Modeler – notação de Barker:**



Figura 29 – Modelo lógico da entidade funcionário  
Fonte: Elaborado pela autora (2014)

Veja que os atributos que eram tratados como específicos na solução envolvendo especialização/generalização são opcionais quando implementados em uma única entidade, uma vez que não se aplicam a todos os funcionários.

Nesse caso, precisaríamos ainda de um **atributo que caracterize a categoria funcional do empregado**, para que possamos preencher os atributos correspondentes à categoria funcional do empregado.

**Por exemplo:** Se a categoria funcional for “MEDICO”, devemos preencher o atributo “nr\_crm” (número do crm), se a categoria funcional for “ENGENHEIRO”, devemos preencher o atributo “nr\_crea” (número do CREA).

**Exemplo:** criando as três entidades e replicando os atributos em cada uma delas, para a situação proposta no exemplo 2.

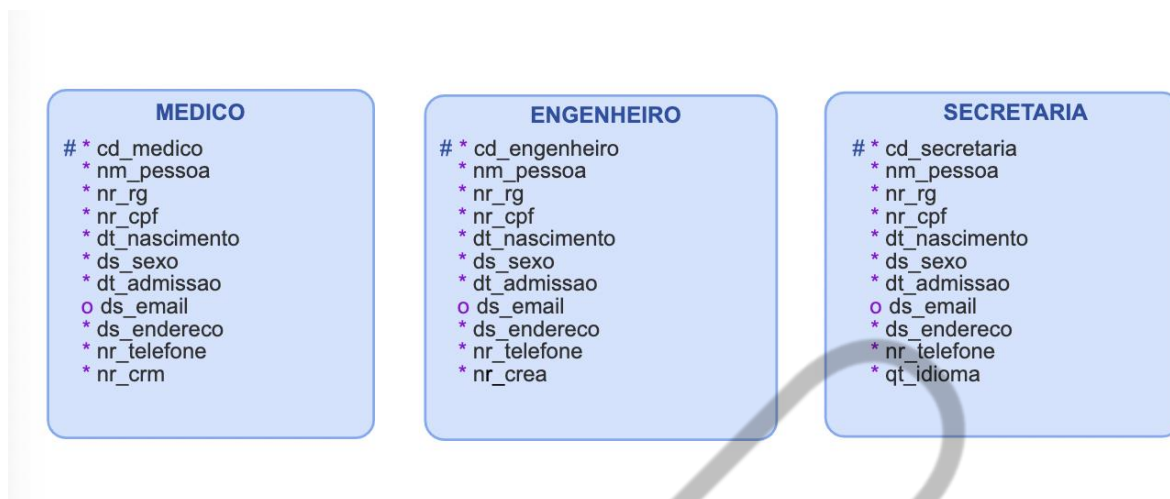
**SQL Data Modeler – Notação de Barker:**

Figura 30 – Modelo lógico das entidades médico, engenheiro e secretária  
 Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

Poderíamos ainda implementar a solução a partir de três entidades distintas. Quando trabalhamos dessa forma, todos os atributos são replicados a cada entidade.

Observamos que os atributos ora tratados como específicos (especialização/generalização), ora tratados como opcionais (quando temos uma única entidade), nessa implementação são atributos mandatórios, uma vez que cada entidade tem um papel específico dentro do contexto de negócio aplicado.

### 1.2.2 Regras para determinar a chave primária nas entidades especializadas

Para que uma entidade seja considerada especialização de outra, é necessário que ela herde o identificador da entidade genérica.

Como temos um relacionamento 1:1 IDENTIFICADO, a chave primária da entidade especializada é a chave estrangeira, proveniente do relacionamento entre “PESSOA” e “PESSOA\_FISICA”; “PESSOA” e “PESSOA\_JURIDICA”.

Vamos visualizar em termos de ocorrências para um melhor entendimento da solução proposta (Figura “Exemplos de ocorrências para as futuras tabelas pessoa, pessoa física e pessoa jurídica”).





Figura 31 – Exemplos de ocorrências para as futuras tabelas pessoa, pessoa física e pessoa jurídica

Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

### 1.2.3 Classificação ou Restrições em uma especialização/generalização

**Classificação:** Total ou Parcial.

**Total:** Para cada ocorrência da entidade genérica (SUPERCLASSE) existe sempre uma ocorrência em uma das entidades especializadas (SUBCLASSE).

**Exemplo:** Toda ocorrência da entidade “PESSOA” corresponde a uma ocorrência em uma das especializações (entidades “Pessoa Física” ou “Pessoa Jurídica”).

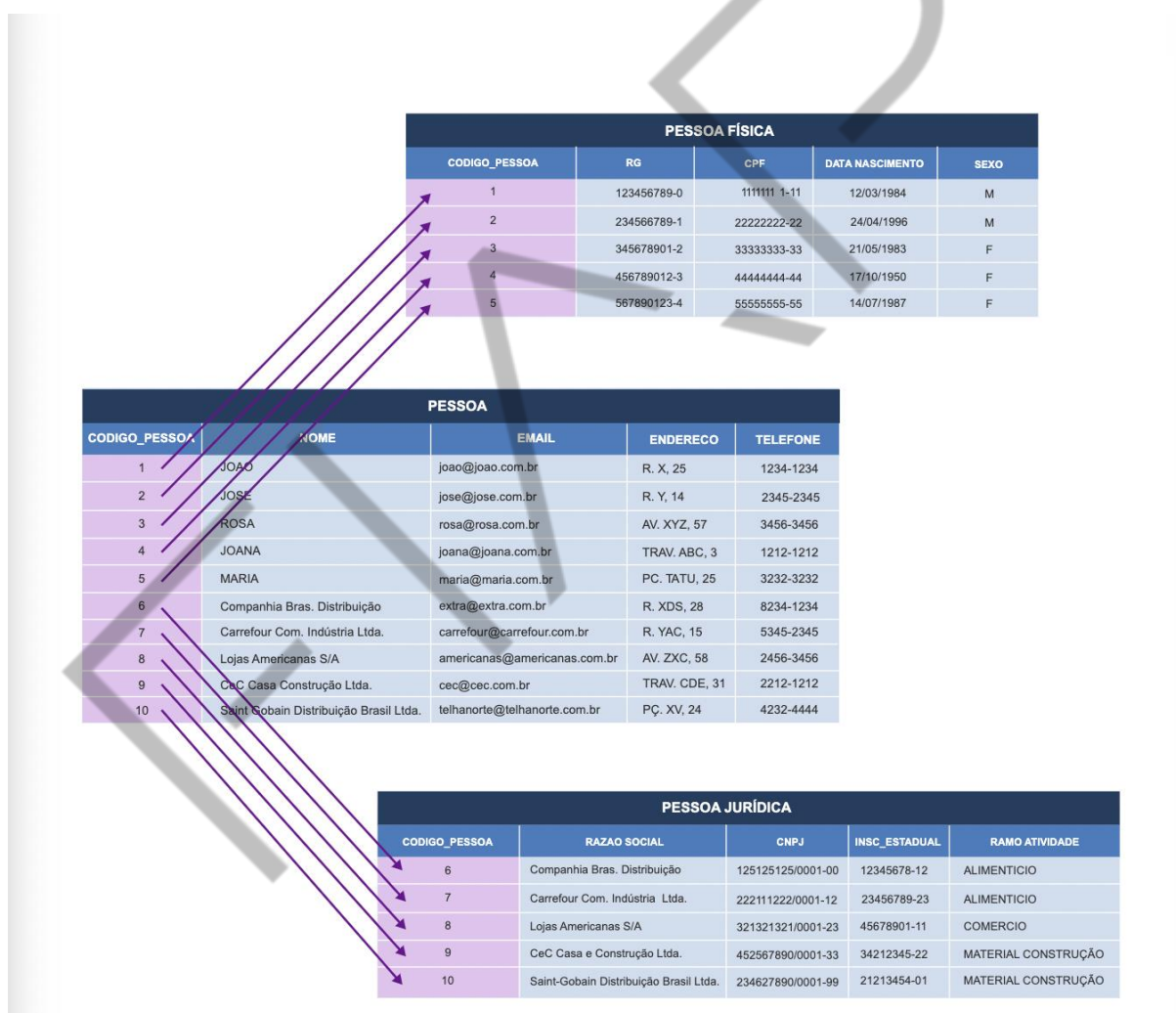


Figura 32 – Exemplos de ocorrências para as futuras tabelas pessoa, pessoa física e pessoa jurídica (2)

Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

**Parcial:** Nem toda ocorrência da entidade genérica (SUPERCLASSE) possui uma ocorrência correspondente em uma entidade especializada (SUBCLASSE).

**Exemplo:** Nem todas as ocorrências da entidade “Funcionário” possuem ocorrências correspondentes nas entidades especializadas (nem todo funcionário é Engenheiro, Médico ou Secretária).

Quando há uma especialização/generalização parcial, podemos ter um atributo na entidade genérica que identifica o tipo de ocorrência, no caso deste exemplo o atributo seria a categoria funcional. Mas não se faz necessário uma vez que conseguimos recuperar informações em múltiplas entidades (tabelas) com os recursos dos comandos SQL que serão vistos mais à frente.

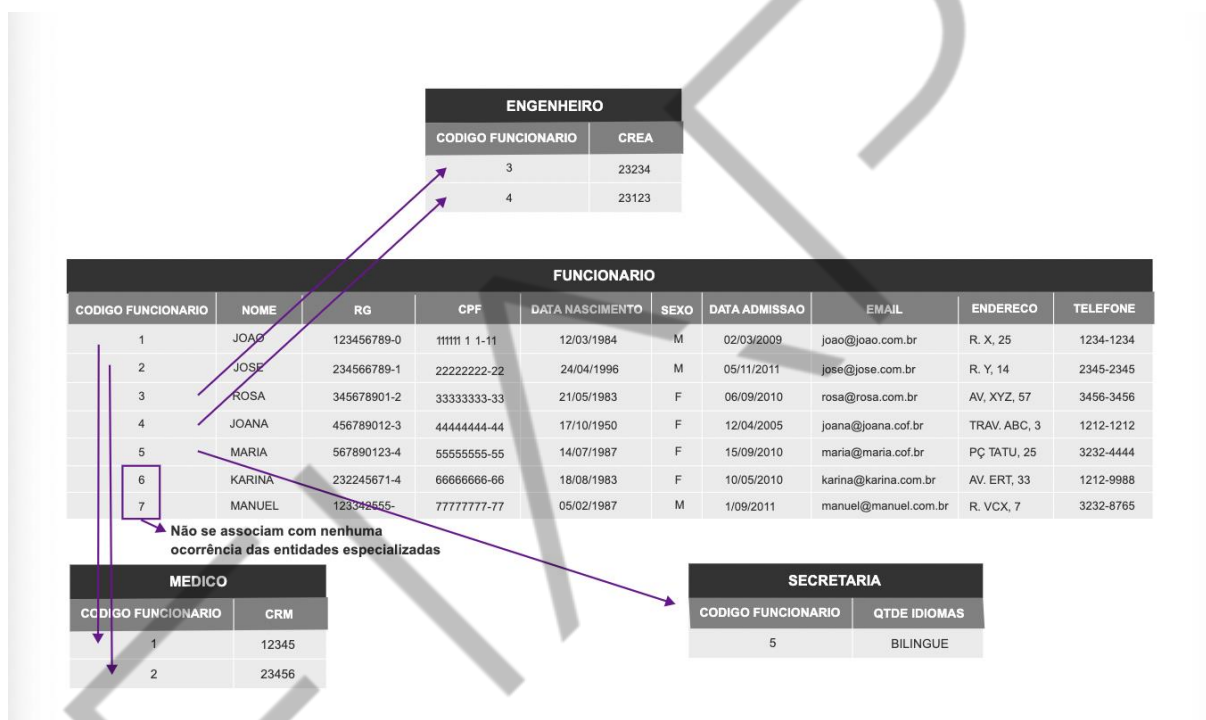


Figura 33 – Exemplos de ocorrências para as futuras tabelas funcionário, secretária, médico e engenheiro

Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

### Classificação: Exclusiva ou Compartilhada

**Exclusiva:** Uma ocorrência da entidade genérica é especializada no máximo uma vez.

**Exemplo:** Uma ocorrência de “FUNCIONARIO” aparece uma vez somente nas entidades especializadas (“ENGENHEIRO”, “MEDICO” ou “SECRETARIA”), já que um funcionário ou é médico, ou é engenheiro ou é secretária.

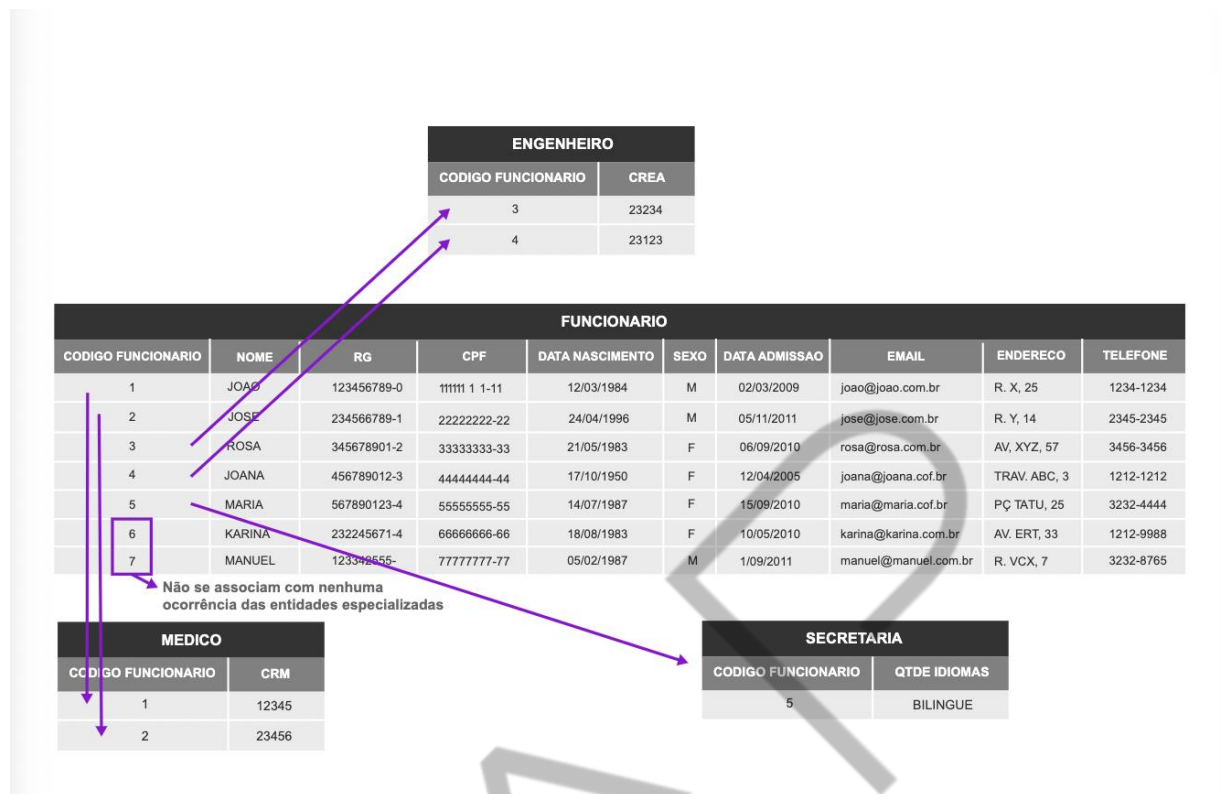


Figura 34 – Exemplos de ocorrências para as futuras tabelas funcionário, secretária, médico e engenheiro (2)  
 Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

**COMPARTILHADA:** Uma ocorrência da entidade genérica pode ser especializada várias vezes (poderá aparecer em várias entidades especializadas).

**Exemplo:** Uma pessoa pode ser professor e aluno ou funcionário e aluno ao mesmo tempo.

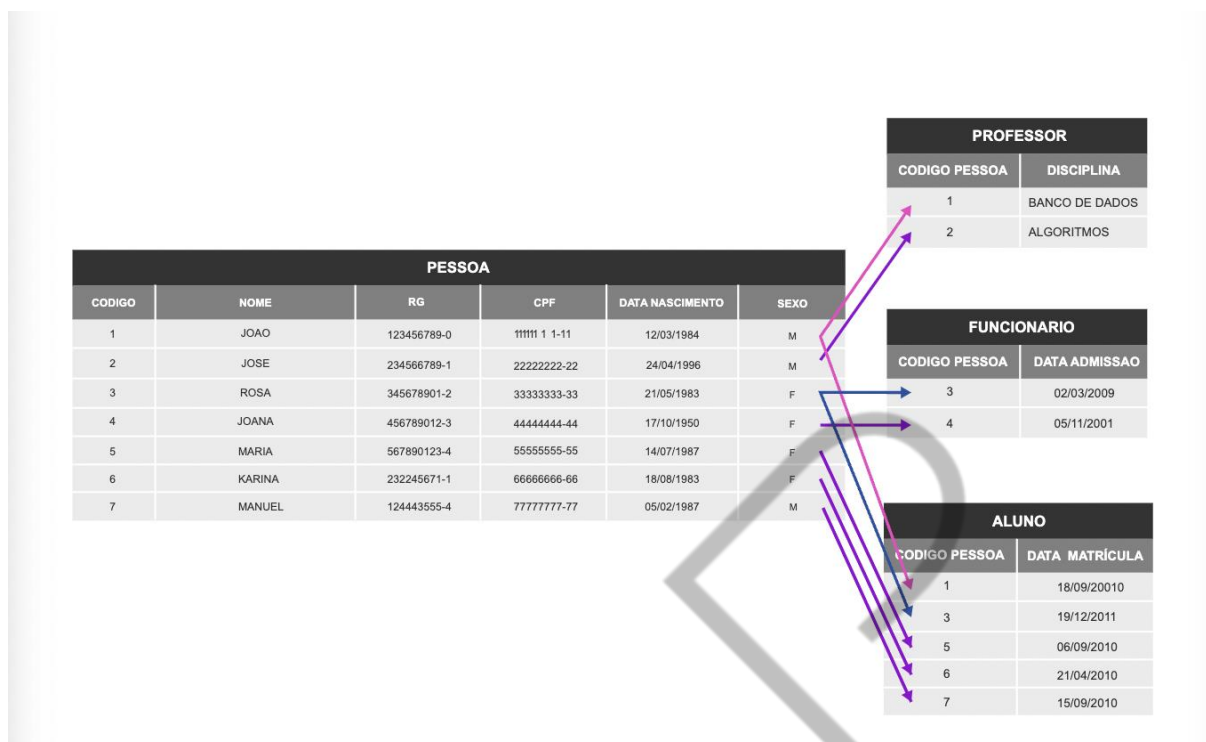


Figura 35 – Exemplos de ocorrências para as futuras tabelas pessoa, professor, funcionário e aluno.

Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

#### 1.2.4 Exemplo de situações em que utilizamos ou não especialização/generalização

##### Situação 1 – Utilizando Especialização/Generalização:

Na situação a seguir, apresentamos as entidades “PESSOA\_FISICA” e “PESSOA\_JURIDICA” desempenhando um mesmo papel, ou seja, ambas as entidades representam “CLIENTE”.

Quando temos entidades distintas desempenhando um mesmo papel, observamos que as ações dessas entidades dentro de um determinado contexto de negócio podem ser as mesmas.

Nesse caso, ambas as entidades possuem a mesma ação, que no exemplo é representada pela realização de “PEDIDO”. Observe que a entidade “PEDIDO” possui as chaves estrangeiras opcionais, pois não é possível um mesmo pedido pertencer a pessoas diferentes.

Se implementado dessa forma, se faz necessário criar alguma regra que valide se uma das chaves estrangeiras é preenchida durante a realização de um pedido,

caso contrário, teremos uma inconsistência, ou seja, poderá existir um pedido que não pertence a nenhuma pessoa, uma vez que a chave estrangeira é um atributo opcional.

### SQL Data Modeler – Notação da Engenharia da Informação:

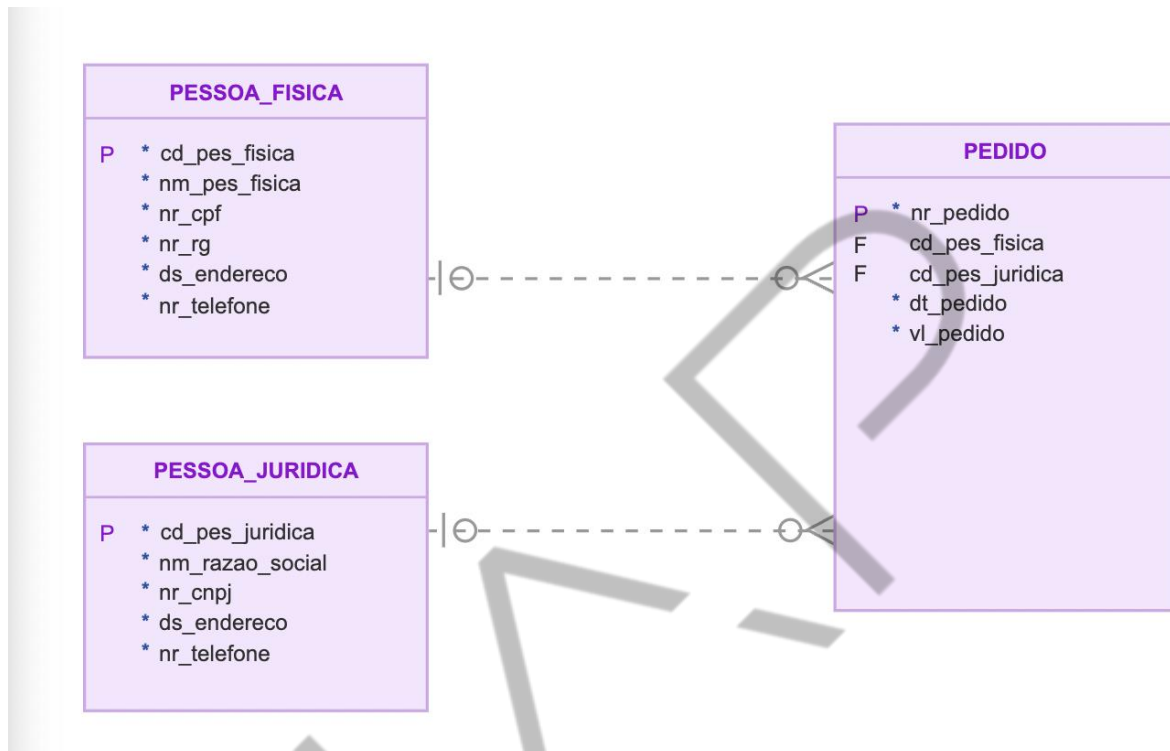


Figura 36 – Modelo lógico entre pedido, pessoa física e pessoa jurídica  
Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

Para resolver essa situação, podemos utilizar o conceito de especialização/generalização.

No exemplo a seguir, representamos a mesma situação utilizando este conceito. Observe que criamos uma entidade genérica chamada “CLIENTE” que representa todos os clientes (pessoa física ou jurídica). Essa entidade genérica será associada com todas as entidades, nela, pessoa física e pessoa jurídica possuirão ações comuns. Neste exemplo, a ação comum é realizar “PEDIDO”. Veja que, quando aplicamos o conceito de especialização/generalização, deixamos de ter o problema da chave estrangeira opcional, em função da entidade genérica “CLIENTE” se associar com a entidade “PEDIDO”.

### SQL Data Modeler – Notação da Engenharia da Informação:



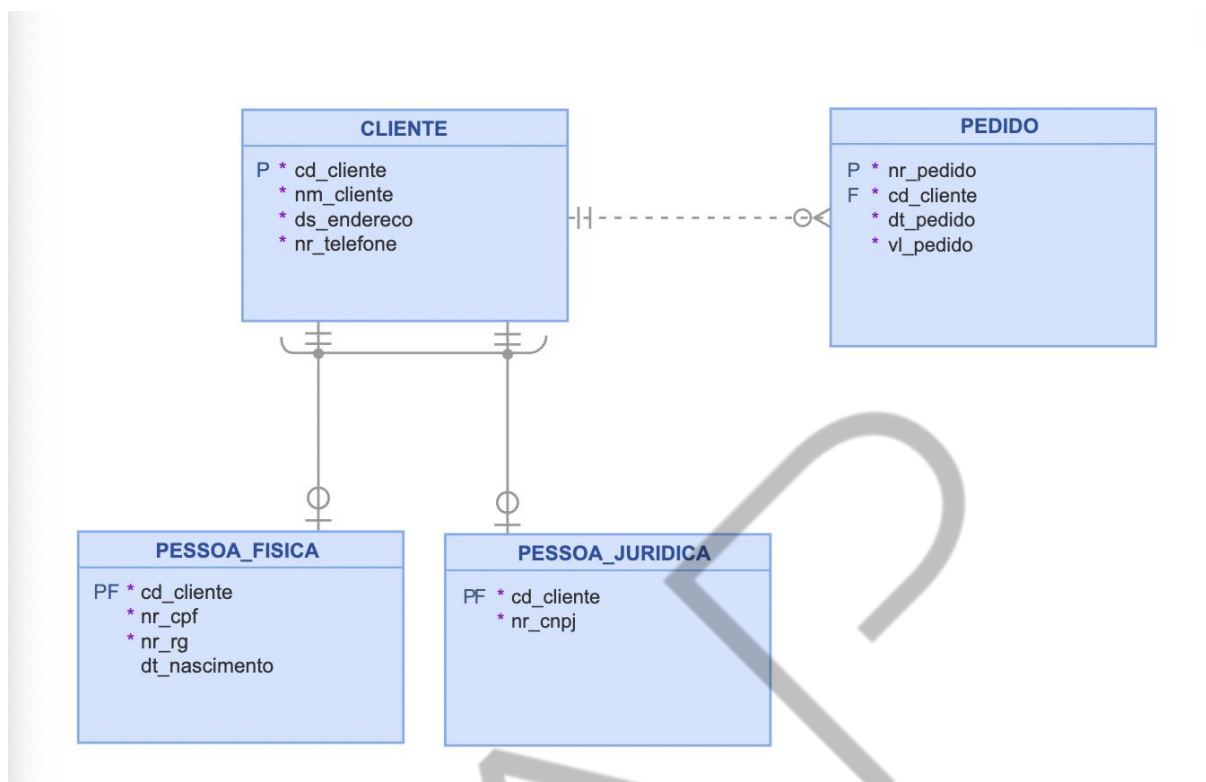


Figura 37 – Modelo lógico entre pedido, cliente, pessoa física e pessoa jurídica  
 Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

As entidades especializadas “PESSOA\_FISICA” e “PESSOA\_JURIDICA” se associam apenas com entidades, suas ações são particulares, ou seja, se aplicam individualmente para pessoa física ou pessoa jurídica.

### **Situação 2 – Quando não é necessário utilizar Especialização/Generalização:**

Na situação a seguir, a entidade “PESSOA\_FISICA” representa um “CLIENTE” e a entidade “PESSOA\_JURIDICA” representa um “FORNECEDOR”, como temos entidades desempenhando papéis diferentes e, conseqüentemente, ações diferentes, não é necessário aplicar o conceito de especialização/generalização.

Caso as entidades desempenhassem o mesmo papel (tanto pessoa física quanto jurídica são clientes ou fornecedores), teriam as mesmas ações (realizar pedidos e/ou fornecer produtos), e nesse caso, seria indicada a utilização da especialização/generalização.

### **SQL Data Modeler – Notação de Barker:**

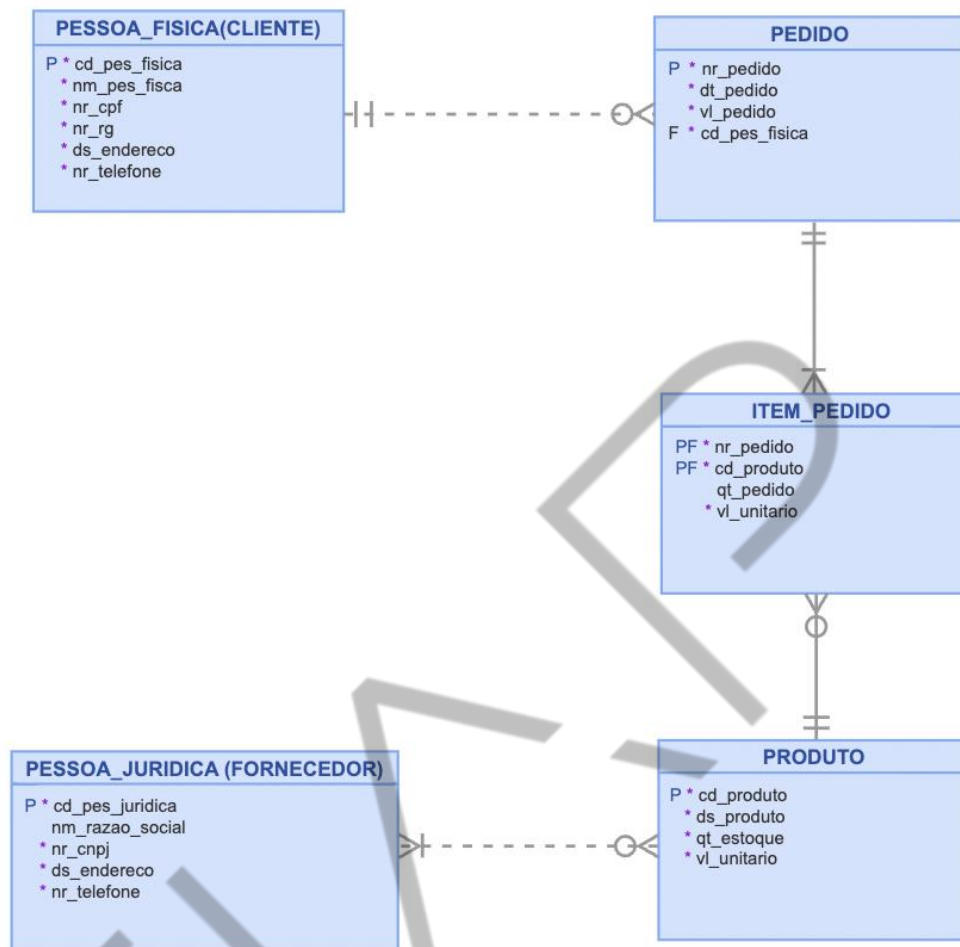


Figura 38 – Modelo lógico formalizando algumas entidades  
 Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

### 1.3 Sobre Autorrelacionamento

O modelo de dados pode e deve passar por um processo de refinamento. Observamos algumas situações particulares, que chamamos de extensões, cuja função é de possibilitar a correção destas particularidades para que o modelo de dados possa ser posteriormente implementado de forma física, por meio de um SGBD.

Uma dessas extensões é conhecida como relacionamento recursivo ou autorrelacionamento e acontece quando as ocorrências de uma entidade se relacionam com ela própria.



Decorre quando as ocorrências de uma **mesma entidade** desempenham papéis diferentes dentro de um contexto de negócio ou para representar algum tipo de hierarquia.

Podemos citar os seguintes exemplos:

- Funcionário que desempenha o papel de **supervisor** de algum departamento.
- Funcionário que desempenha o papel de **gerente** de algum departamento.

Observe que, nas duas situações exemplificadas, temos a ideia de hierarquia.

O relacionamento recursivo evita a criação de entidades idênticas em um modelo, o que difere uma da outra é o papel, imagine os atributos de uma entidade “FUNCIONARIO” e de uma entidade “COORDENADOR”.

Exemplo de alguns atributos da entidade “FUNCIONARIO”:

Número de matrícula

Nome completo

RG

CPF

Data de Nascimento

Endereço

Telefone

Data de Admissão

Exemplo de alguns atributos da entidade “COORDENADOR”:

Número de matrícula

Nome completo

RG

CPF

Data de Nascimento

Endereço

Telefone

Data de Admissão

Temos basicamente os mesmos atributos, o que difere é que a entidade “COORDENADOR” é um papel representado por apenas alguns “FUNCIONÁRIOS” da empresa.

Observe que não faz sentido manter duas entidades com as mesmas características em um modelo, imagine o momento de referenciar um coordenador, que também é um funcionário, ele teria duas identificações diferentes na empresa, por desempenhar um papel específico. Qual identificação seria utilizada para emitir um demonstrativo de pagamento ou para associá-lo a um vale refeição e, quando esse funcionário deixasse de ser gerente, teríamos, no mínimo, que verificar quais identificações e onde estavam sendo utilizadas dentro da estrutura de armazenamento.

Além disso, o coordenador também conta com um gerente com as mesmas características. E o superior desse gerente? E se todos fossem representados no modelo de dados? Com essa abordagem, teríamos várias entidades iguais relacionadas entre si.

Isso poderia provocar algum tipo de inconsistência em relação aos dados armazenados na estrutura de banco de dados, pois ambas as identificações estariam corretas, até que uma delas deixasse de ser ativa.

### 1.3.1 Exemplificando relacionamentos recursivos ou autorrelacionamento

#### **Visão geral da necessidade de um relacionamento recursivo:**

Usamos o relacionamento recursivo em modelos nos quais observamos o relacionamento entre entidades, existindo níveis hierárquicos entre elas. Cada nível representa uma entidade no modelo, conforme situação abaixo:

Exemplo dos níveis hierárquicos em um modelo sem autorrelacionamento: cada nível representa uma entidade, aqui: “COORDENADOR” e “FUNCIONARIO” (Figura “Modelo lógico com as entidades coordenador e funcionário”).

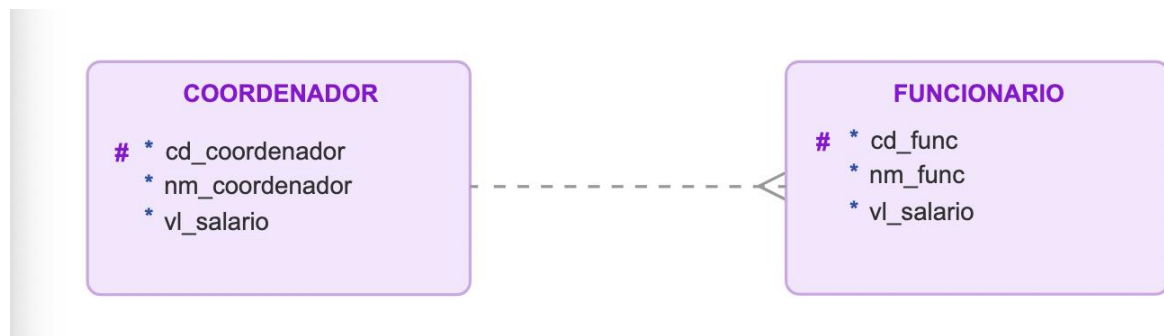


Figura 39 – Modelo lógico com as entidades coordenador e funcionário  
 Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

Analisando o relacionamento em termos de ocorrências (Figura “Exemplos de ocorrências das futuras tabelas coordenador e funcionário”).

COORDENADOR		
CÓDIGO COORDENADOR	NOME	SALÁRIO
123	LAURA	R\$ 4.500,00
124	THABATA	R\$ 3.400,00
125	THAIS	R\$ 4.900,00
126	ANDRE	R\$ 3.800,00
127	KATIA	R\$ 5.100,00

FUNCIONARIO			
CÓDIGO COORDENADOR	CÓDIGO FUNCIONÁRIO	NOME	SALÁRIO
123	1234	JOAO	R\$ 2.300,00
123	1235	JOSE	R\$ 3.200,00
nulo	1236	ROSA	R\$ 2.100,00
124	1237	JOANA	R\$ 1.950,00
127	1238	MARIA	R\$ 3.200,00
127	1239	KARINA	R\$ 2.890,00
126	1240	MANUEL	R\$ 2.789,00
126	1241	CAROLINA	R\$ 2.120,00
nulo	1242	PEDRO	R\$ 1.700,00
nulo	1243	ROBERTO	R\$ 1.800,00

Figura 40 – Exemplos de ocorrências das futuras tabelas coordenador e funcionário  
 Fonte: Elaborado pela autora (2014), adaptado por FIAP (2017)

Sem a utilização do autorrelacionamento, observamos a chave estrangeira “código do coordenador”, na entidade “FUNCIONARIO”.

### Considerações importantes:

Observe que os principais atributos de cada uma das entidades são iguais, dentro do contexto EMPRESA, a natureza dessas entidades é semelhante, pois coordenador e funcionário trabalham na empresa, podemos dizer que todos são funcionários da empresa.

Na empresa, temos uma hierarquia que refletirá no modelo de dados.

Devemos considerar que um nível hierárquico poderá deixar de existir e funções podem ser desmembradas. Caso isso ocorra, o modelo de dados deverá ser revisto e modificado, inclusive as aplicações que se utilizam dessa estrutura.

### Alternativas de modelagem:

#### 1) Criação de um relacionamento recursivo na entidade “FUNCIONÁRIO”, representando a hierarquia da empresa:

Nesta forma, o relacionamento é indicado após a inclusão de todos os elementos na entidade a ser relacionada.

Nesta situação, é considerado um momento no tempo, em que um funcionário é subordinado a um coordenador, não considerando alterações funcionais ao longo do tempo.

#### Dessa forma, apresentamos uma associação recursiva de 1:n:

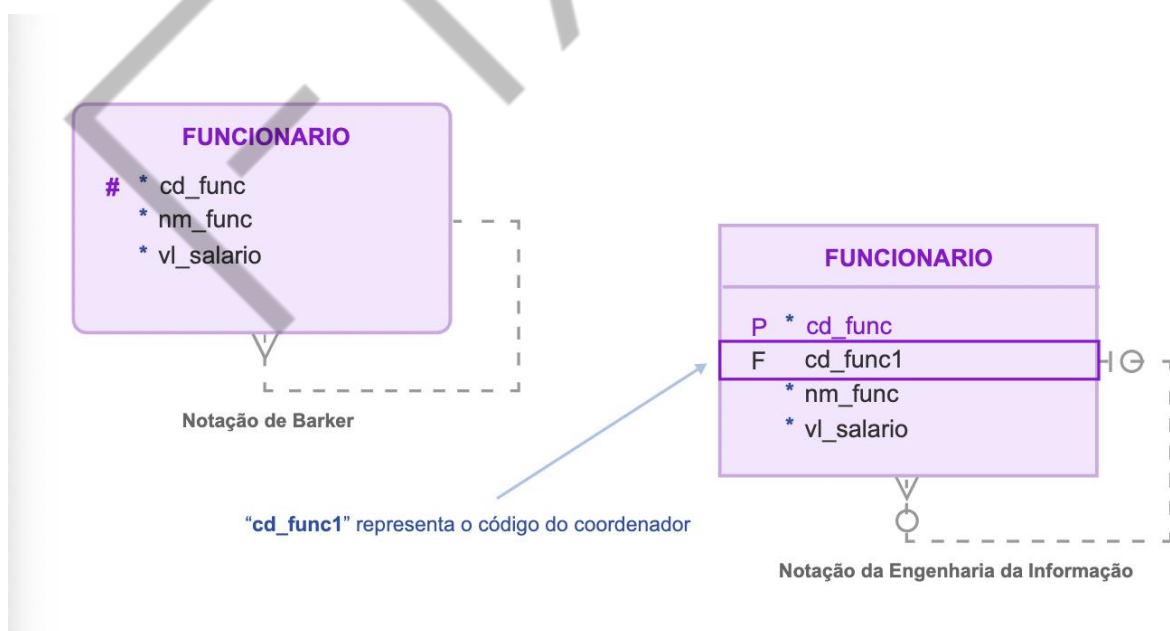


Figura 41 – Modelo lógico representando o autorrelacionamento da entidade funcionário  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Observe que, quando temos um autorrelacionamento, temos a **chave estrangeira na mesma entidade**. No exemplo, a chave estrangeira “**cd\_func1**” faz referência à chave primária “**cd\_func**”.

Para facilitar o entendimento, podemos renomear a chave estrangeira “**cd\_func1**” para “**cd\_coordenador**”, por exemplo.

Analisando o relacionamento em termos de ocorrências. (Figura “Exemplos de ocorrências da futura tabela funcionário com autorrelacionamento”).

FUNCIONARIO			
CÓDIGO COORDENADOR	CÓDIGO FUNCIONÁRIO	NOME	SALÁRIO
1234	1244	JOAO	R\$ 2.300,00
1235	1244	JOSE	R\$ 3.200,00
1236	nulo	ROSA	R\$ 2.100,00
1237	1245	JOANA	R\$ 1.950,00
1238	1248	MARIA	R\$ 3.200,00
1239	1248	KARINA	R\$ 2.890,00
1240	1247	MANUEL	R\$ 2.789,00
1241	1247	CAROLINA	R\$ 2.120,00
1242	nulo	PEDRO	R\$ 1.700,00
1243	nulo	ROBERTO	R\$ 1.800,00
1244	nulo	LAURA	R\$ 4.500,00
1245	nulo	THABATA	R\$ 3.400,00
1246	nulo	THAIS	R\$ 4.900,00
1247	nulo	ANDRE	R\$ 3.800,00
1248	nulo	KATIA	R\$ 5.100,00

Figura 42 – Exemplos de ocorrências da futura tabela funcionário com autorrelacionamento  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

## 2) Considerando que haverá, ao longo do tempo, mudanças de subordinação:

### Interpretação:

Um funcionário pode ser subordinado a vários funcionários (coordenador) diferentes ao longo do tempo.

Um coordenador pode ter vários subordinados (funcionários) ao longo do tempo.

Teríamos assim uma **associação recursiva m:n** (Figura “lógico representando o autorrelacionamento da entidade funcionário (2)”).

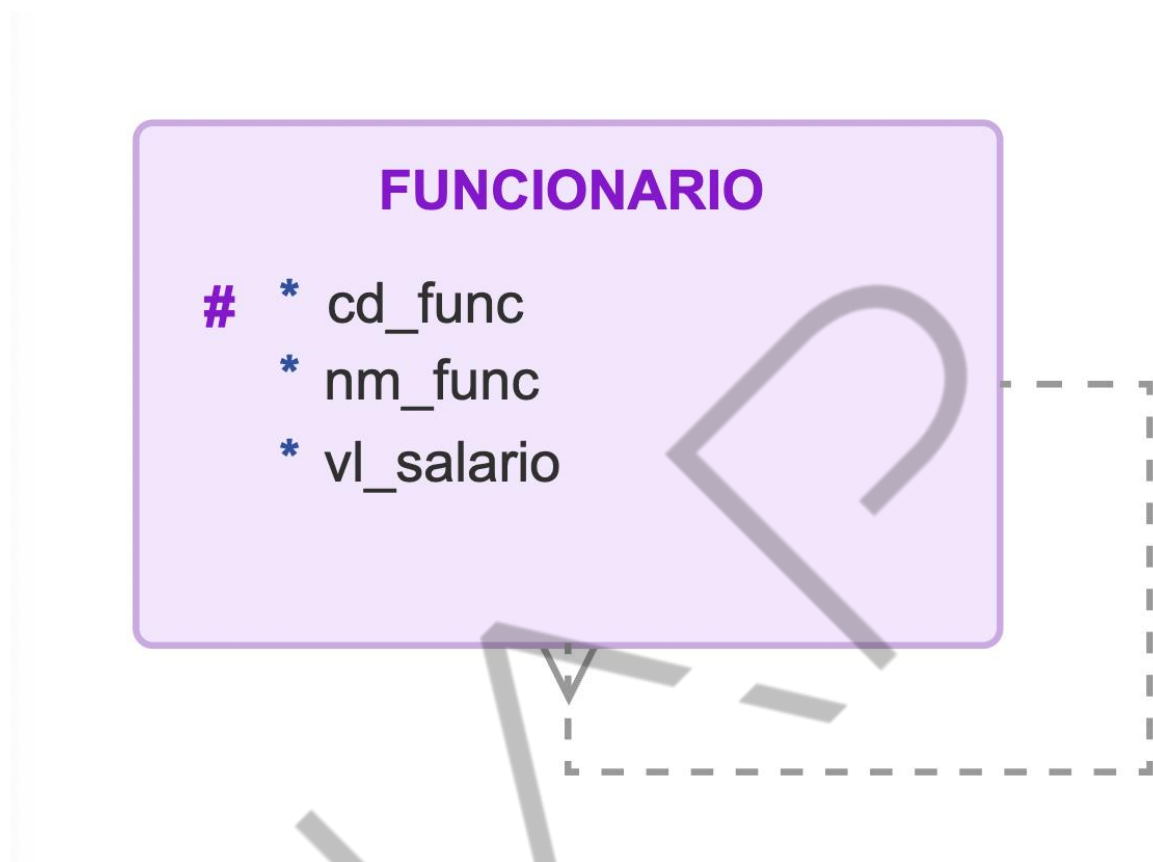


Figura 43 – Modelo lógico representando o autorrelacionamento da entidade funcionário (2)  
Fonte: Elaborado pela autora (2014)

Considerando a associação recursiva m:n, teremos que **criar uma entidade** relacionando “FUNCIONÁRIOS” e “SUBORDINADOS”. Dessa maneira, flexibilizamos a estrutura, podendo assim **indicar data de início e término de um vínculo entre funcionários – coordenador x subordinado** (figura “Modelo lógico das entidades funcionário e relacionamento\_funcionario, notação de Barker”).

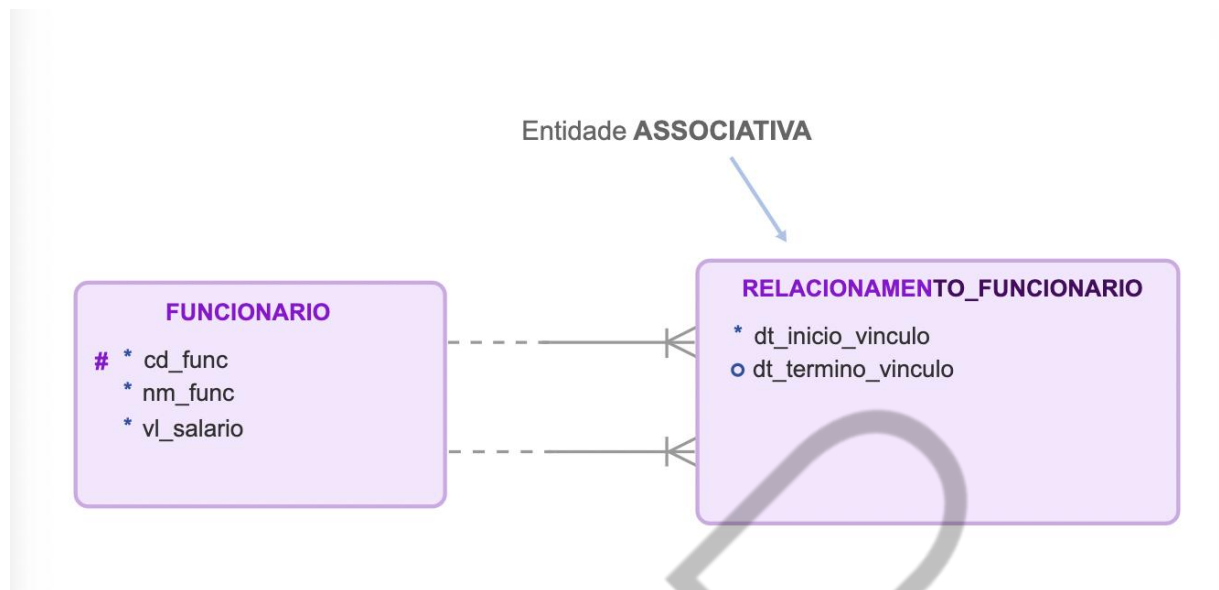


Figura 44 – Modelo lógico das entidades funcionário e relacionamento\_funcionario, notação de Barker  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

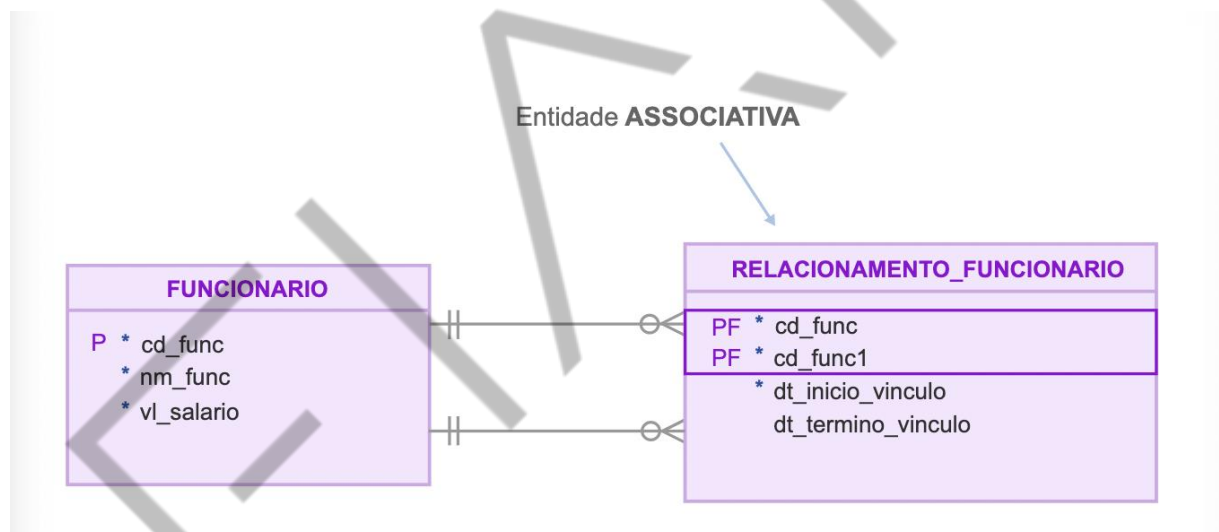


Figura 45 – Modelo lógico das entidades funcionário e relacionamento\_funcionario, notação da Engenharia da Informação  
Fonte: Elaborado pela autora, adaptado por FIAP (2017)

Analisando o relacionamento em termos de ocorrências (Figura “Exemplos de relacionamentos entre funcionário e relacionamento\_funcionario”).



Figura 46 – Exemplos de relacionamentos entre funcionário e relacionamento\_funcionario  
 Fonte: Elaborado pela autora (2014)

Exemplos de situações em que podemos aplicar relacionamento recursivos:

- Representação de pessoas casadas em regimes monogâmicos ou não.
- Indicações de pessoas, livros, filmes.
- Indicações de amigos (rede de amizades).
- Composição de materiais (itens formados por outros itens: veículos automotores compostos por: carro – motor, direção, câmbio; moto – carburador, velas, platinado).



## REFERÊNCIAS

ELMASRI, R.; NAVATHE, S.B. **Sistemas de Banco de Dados**: Fundamentos e Aplicações. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

HEUSER, C.A. **Projeto de Banco de Dados**. Série Livros Didáticos. v. 4. 6. ed. Rio Grande do Sul: Bookman, 2009.

MACHADO, F. N.R. **Banco de Dados** - Projeto e Implementação. 2. ed. São Paulo: Érica, 2008.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de Dados**: Implementação em SQL, PL/SQL e Oracle 11g. São Paulo: Pearson Education do Brasil, 2013.

SILBERSCHATZ, A; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Campus, 2006.